

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601

*Mathématiques, Télécommunications, Informatique, Signal, Systèmes,
Électronique*

Spécialité : *INFO*

Par

Antoine CHAFFIN

Multimodal Misinformation Detection

Overcoming the Training Data Collection Challenge Through Data Generation

Thèse présentée et soutenue à Rennes, le 14 novembre 2023

Unité de recherche : IRISA

Thèse N° : 2020/0501

Rapporteurs avant soutenance :

Olivier FERRET Chercheur senior, CEA LIST, LASTI, Gif-Sur-Yvette
Benoit FAVRE Professeur, Université Aix-Marseille, LIS, Marseille

Composition du Jury :

Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition du jury doit être revue pour s'assurer qu'elle est conforme et devra être répercutée sur la couverture de thèse

Examineurs :	Damien LOLIVE	Professeur, Université de Rennes, IRISA, Lannion
	Olivier FERRET	Chercheur senior, CEA LIST, LASTI, Gif-Sur-Yvette
	Benoit FAVRE	Professeur, Université Aix-Marseille, LIS, Marseille
	Claire GARDENT	Directrice de recherche, CNRS, LORIA, Vandoeuvre-lès-Nancy
Dir. de thèse :	Ewa KIJAK	Maitre de conférence, Université de Rennes, IRISA, Rennes
Co-dir. de thèse :	Vincent CLAVEAU	Chargé de recherche, CNRS, DGA, Bruz

Invité(s) :

Sylvain LAMPRIER Professeur, Université d'Angers, LERIA, Angers
Vivien CHAPPELIER Chercheur senior, IMATAG, Rennes

REMERCIEMENTS

Je tiens à remercier mes parents pour leur soutien inconditionnel tout au long de mes études, sans lequel je n'aurais pas pu en arriver là. Je les remercie également pour avoir rendu les confinements durant la thèse, une période complexe pour beaucoup, un véritable moment agréable que je n'ai pas eu à subir. Pour cette raison, je remercie également ma conjointe Mathilde, ainsi que pour son soutien tout au long de la thèse, dans les bons moments comme dans les plus compliqués.

Je remercie toutes les personnes qui m'ont permis de découvrir la recherche et de progresser dans ce domaine. Tout d'abord mes encadrants de thèse, Vincent et Ewa. Ils m'ont accompagné durant la thèse, mais également durant mon stage de fin d'études, moment où j'ai décidé de continuer en thèse. Je remercie également tous les chercheurs et doctorants de l'équipe LinkMedia avec lesquels j'ai pu partager de bons moments lors de mes passages (sporadiques) au laboratoire.

Ces mêmes remerciements s'adressent à l'équipe MLIA à l'université de la Sorbonne, qui m'a accueilli chaleureusement pendant un mois. Je tiens à remercier particulièrement Sylvain, Benjamin et Thomas pour cette collaboration qui m'a énormément appris.

Ainsi, je remercie également toute l'équipe d'IMATAG pour m'avoir accueilli pour cette thèse, notamment l'équipe recherche, Vivien, Vedran et Gautier pour nos nombreuses discussions très instructives et la bonne humeur ambiante de cette équipe.

Je tiens à remercier particulièrement l'équipe de doctorants soudée depuis le double diplôme : Julien (mon cher co-auteur), Taha, Enzo, Gaël, Olivier et William. Pouvoir discuter de sciences et de sujets divers ainsi que rire avec des amis qui traversent eux aussi cette épreuve a une valeur inestimable. Ces mêmes remerciements s'adressent à Paul, qui a supporté des heures durant mes différentes digressions scientifiques et qui m'a permis de maturer mes idées et mes connaissances.

Enfin, je tiens à remercier NVIDIA pour leurs cartes graphiques et leur monopole sur ce secteur.

RÉSUMÉ EN FRANÇAIS

La désinformation, une menace contemporaine La croissance des réseaux sociaux a créé un essor sans précédent de la liberté d'expression en permettant à tous de partager ses opinions et ses idées. Cette croissance a été accompagnée par une augmentation de la quantité d'informations disponibles sur ces réseaux, poussant les gens à s'informer de plus en plus sur ces plateformes plutôt que sur les médias traditionnels¹ [222].

Cependant, la qualité de l'information sur ces réseaux est bien inférieure à celle des médias traditionnels. En effet, chaque utilisateur peut transmettre une information, sans assurance sur ses intentions ainsi que son expertise du domaine et sans avoir à être tenu responsable de ses publications. Cela mène à la publication de fausses informations qui vont se propager au travers du réseau à une très grande vitesse [359]. Ces fausses informations peuvent avoir des conséquences considérables, allant de menaces envers la démocratie via la manipulation lors d'élections [43, 423, 106] à des impacts sur le système monétaire², notamment via la manipulation du marché boursier³. Plus grave encore, les fausses informations peuvent mener à la mort de personnes, par exemple lors d'incidents causés par la propagation de fausses informations (e.g., le Pizzagate⁴ et des émeutes en Inde⁵) ou par la propagation de fausses informations sur la santé⁶ causant des défiances sur les politiques de santé [124, 153]. Malheureusement, l'activité sur les réseaux sociaux est si forte⁷ qu'il est impossible de vérifier chaque information manuellement.

Les utilisateurs des réseaux sociaux sont donc exposés continuellement à de la désinformation, à laquelle ils sont particulièrement vulnérables à cause de différents biais [39, 245, 108, 193], en plus de ne pas réussir à la détecter efficacement [285, 420]. Enfin, il est difficile

1. <https://www.pewresearch.org/journalism/2018/09/10/news-use-across-social-media-platforms-2018/>

2. <https://www.zdnet.com/article/online-fake-news-costing-us-78-billion-globally-each-year/>

3. <https://www.forbes.com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/>

4. <https://time.com/4590255/pizzagate-fake-news-what-to-know/>

5. <https://asiatimes.com/2019/10/indias-fake-news-problem-is-killing-real-people/>

6. <https://www.who.int/news-room/feature-stories/detail/fighting-misinformation-in-the-time-of-covid-19-one-click-at-a-time>

7. <https://datareportal.com/social-media-users>

de corriger des croyances une fois qu'une information a gagné notre confiance [83, 31]. Il est donc nécessaire de développer des outils automatiques pour détecter les fausses informations et les marquer comme telles le plus rapidement possible. Des outils de détections automatiques sont donc nécessaires afin d'aider les organismes de vérification manuelle de l'information.

Bien que l'information soit majoritairement transmise sous la forme de texte, elle est souvent accompagnée d'images afin de rendre la publication plus attrayante [102], maximiser les interactions ainsi que la diffusion⁸ [404] et augmenter les convictions du lecteur envers l'information [131, 103, 328, 199, 8, 404, 138].

De plus, l'image est un vecteur de propagation de désinformation. Au-delà de la manipulation pure du contenu visuel [129, 136, 93], le cas le plus répandu est celui de la réutilisation de contenu⁹, c'est-à-dire réutiliser une image ancienne en la détournant de son contexte original. Ce type de désinformation est très populaire de par son faible coût : il suffit à l'attaquant de choisir une image qui semble supporter son propos. De plus, comme l'image n'a pas été modifiée, il n'y a aucune trace de manipulation [62] qui pourrait rendre le lecteur suspicieux ou faciliter la détection.

La détection automatique de désinformation Diverses approches ont été proposées pour détecter automatiquement la désinformation. Certaines se basent sur les informations autour du contenu lui-même, comme la façon dont il se propage [415, 37, 313, 187] et les profils des utilisateurs impliqués [55, 314, 287, 357, 322, 26]. Elles ne sont donc malheureusement pas applicables pour les cas où ces informations ne sont pas disponibles ou lorsque l'information est encore dans un stade précoce et ne s'est pas beaucoup propagée. Or, c'est au début de la propagation, avant que beaucoup d'utilisateurs y aient été exposés, qu'il est le plus intéressant d'agir.

D'autres approches se basent directement sur le contenu, au travers de l'extraction de caractéristiques statistiques et sémantiques. Les textes trompeurs étant écrits différemment [9, 422], il est possible de les détecter en se basant sur des caractéristiques stylistiques et linguistiques [422, 55, 265, 223, 101, 259]. Des approches similaires existent pour détecter des biais dans les images [164, 41, 32, 416, 52]. Certaines approches basées sur le contenu étudient la sémantique de ce dernier [372, 419, 320, 64, 55, 173]. Ce sont ces approches qui

8. <https://www.fastcompany.com/3022116/whattwitters-expanded-images-mean-for-clicks-retweets-andfavorites/>

9. <https://theconversation.com/out-of-context-photos-are-a-powerful-low-tech-form-of-misinformation-129959/>

se rapprochent le plus du *fact-checking* fait par les journalistes, en analysant et vérifiant les informations contenues dans les publications. Certaines études utilisent uniquement le texte des publications, sans prendre en compte l'information contenue dans l'image, et de manière plus critique, l'interaction entre le texte et l'image. Les premières études multimodales concaténaient simplement les représentations de chaque modalité [372, 320, 321], négligeant ainsi les interactions complexes entre les deux modalités. Afin de mieux modéliser les relations complexes, certaines études s'intéressent à la cohérence entre ces dernières [419, 390, 118] en utilisant des représentations plus ou moins complexes.

Se baser uniquement sur la sémantique contenue dans la publication permet de détecter des manipulations simples, comme l'échange aléatoire d'une image et de sa description [157]. Cependant, cela ne permet pas de détecter des manipulations plus subtiles, comme le remplacement d'une entité par une entité similaire du même type (e.g., une personne par une autre personne) [290] ou le remplacement par une image sémantiquement similaire [215]. Ainsi, certaines approches utilisent des connaissances extérieures afin de vérifier que la publication est conforme à des faits connus et vérifiés.

Les approches qui se basent sur le contenu de la publication ainsi que celles qui utilisent des connaissances extérieures reposent sur des espaces de représentation où les différentes modalités peuvent être examinées et comparées. Ce sont ces représentations qui permettent de récupérer les informations pertinentes dans les bases de connaissances externes et de les comparer avec les informations contenues dans la publication. **Ces représentations sont donc cruciales pour la performance des approches.**

Les Transformers [353] ont poussé la qualité de ces représentations à un tout autre niveau grâce à la grande capacité de leur mécanisme d'attention leur permettant de capturer des relations complexes entre les différents éléments d'une séquence. Ce mécanisme est mis à contribution lors d'une grande phase de pré-entraînement dans laquelle le modèle va apprendre des espaces de représentation en utilisant une immense quantité de données non annotées via des tâches d'auto-supervision. Ces représentations peuvent ensuite être utilisées pour résoudre des tâches spécifiques en les adaptant à la tâche en question via un entraînement supervisé. Elles sont donc très générales et peuvent être utilisées pour de nombreuses tâches différentes.

Créer un jeu de données adapté Cependant, même en utilisant un modèle pré-entraîné, il est nécessaire de l'adapter pour la tâche à accomplir en utilisant des données représentatives de cette dernière. Dans le cas de la détection de désinformation, ce jeu de données doit

être composé de vraies informations ainsi que d'exemples de désinformation. **Cependant, la création d'un tel jeu est complexe pour différentes raisons.** Trois approches peuvent être considérées pour le créer : collecter les données, faire des échanges entre les modalités afin de créer des décalages entre le texte et l'image et enfin, générer des exemples synthétiques.

Même si l'on pourrait supposer que l'augmentation de la désinformation couplée à l'activité des réseaux sociaux faciliterait la collecte de données, la tâche reste complexe. Les réseaux de neurones ont besoin d'une quantité de données importante et il n'existe pas de "source fiable" de désinformation (sans considérer la satire [286]). Ainsi, contrairement aux vraies informations qui peuvent être collectées dans les journaux, la collecte de désinformation requiert une étape de fact-checking en amont. Cela limite énormément la quantité de données que l'on peut collecter. Même si les premières études se focalisaient sur le texte [369, 336, 296], un certain nombre de jeux de données composés d'images associées aux textes ont été proposés [315, 238, 165, 40, 163, 430].

Cependant, la différence entre les sources des données crée un biais dans ces jeux. Les vraies informations sont sous la forme d'articles documentés et écrits par des journalistes professionnels, tandis que les exemples de désinformation sont issus de sources diverses et souffrent des biais stylistiques mentionnés précédemment. Ainsi, les modèles peuvent apprendre à différencier les deux sources plutôt que de se focaliser sur le contenu. Cela permet aux modèles qui n'utilisent qu'une des deux modalités d'obtenir des performances supérieures à ce qui serait attendu [148]. Ces modèles sont donc biaisés et apprennent à identifier des caractéristiques qui ne sont pas généralisables, résultant en des performances qui dépendent énormément du jeu de données [44]. La grande capacité des Transformers les rend vulnérables à l'apprentissage de bruit et de biais contenus dans les données qui ne se retrouveront pas forcément dans des données réelles lors de l'utilisation en production. C'est d'autant plus vrai pour les Transformers multimodaux qui vont s'appuyer fortement sur les biais contenus dans une modalité aux dépens de l'autre [110, 292, 50]. Même si ces caractéristiques peuvent avoir un intérêt pratique pour la détection de désinformation, elles sont très spécifiques aux sources des données et ne reflètent pas la véracité de l'information. Elles ne sont donc pas généralisables à tout type de données, par exemple de la désinformation bien écrite ou de la vraie information écrite de manière peu professionnelle.

Enfin, même si un jeu de données suffisamment grand et non biaisé était créé, il serait périmé à l'instant même de sa création de par la nature dynamique de l'information. Les

faits évoluent (e.g., le président actuel des États-Unis) et notre connaissance du monde également. Suivre le rythme de l'information est déjà une tâche complexe, mais l'étape de fact-checking rend le suivi des derniers sujets de désinformation impossible.

Une autre possibilité pour créer ce jeu de données est d'échanger les images des articles professionnels avec d'autres images afin de créer un décalage entre les deux [215, 157]. NewsCLIPPings [215] est un jeu de données créé ainsi en utilisant les excellentes performances de recherche cross-modal de CLIP [270] pour faire les échanges afin de créer de faux couples très convaincants. Cependant, nous avons choisi de privilégier la génération de données synthétiques pour différentes raisons. En effet, l'échange d'images nécessite une image proche, qui peut ne pas exister ou ne pas être à notre disposition. Mais surtout, cette approche repose sur la qualité d'un alignement cross-modal existant afin de sélectionner un bon remplacement. Apprendre à générer des données synthétiques, au contraire, permet d'apprendre des représentations qui sont utiles pour des tâches discriminatives [405, 269, 28, 257, 348, 227, 397, 368, 332, 364]. Ainsi, plutôt que d'utiliser les capacités d'un modèle existant, nous créons un espace de représentation adapté à la tâche et qui pourra servir de point de départ pour un modèle de détection ou pour récupérer des informations pertinentes dans les bases de connaissances.

Dans cette thèse, nous étudions donc la possibilité d'utiliser uniquement les médias certifiés comme source pour le jeu de données d'entraînement via l'utilisation des modèles génératifs. Ces modèles sont entraînés à reproduire la distribution des exemples d'entraînement, ce qui peut ensuite être utilisé pour générer des données qui suivent cette distribution. Cela résout les problèmes introduits précédemment, puisque cela permet de générer un nombre souhaité de documents sur des sujets ne reposant pas sur des faits connus qui ressemblent à de vrais articles. Grover [405] a montré qu'il était possible d'entraîner un modèle de langue à générer des articles qui suivent le format et le style d'un journal souhaité, à propos d'un sujet défini par le titre donné en entrée du modèle. Ainsi, des versions modifiées de titres existants peuvent être données à ce type de modèle afin de générer de faux articles reliés à l'image de l'article original [339]. Les modifications apportées aux titres doivent être suffisamment subtiles, comme le remplacement d'une personne par une autre. Même si cette approche produit des articles très convaincants qui peuvent facilement tromper les humains [405], les textes générés ne suivent pas exactement la distribution des textes collectés. Ces différences statistiques sont facilement détectables par les réseaux de neurones, qui peuvent apprendre à les utiliser pour différencier les vrais articles des faux. **Ainsi, une grande partie de ce manuscrit est consacrée à réduire**

la différence entre ces deux distributions. Veuillez noter que nous avons fait le choix de conserver le titre original de la thèse même si les contributions que nous allons présenter s'éloignent quelque peu de l'objectif initial de détection de désinformation. Cependant, nous avons souhaité replacer ces contributions dans ce contexte applicatif car il constitue la base de notre motivation pour ces différents travaux.

Générer des données d'entraînement Nous commençons ce manuscrit par une étude préliminaire sur **l'utilisation de données textuelles synthétiques en complément ou en substitution des données originales pour des tâches de classifications supervisées** dans le Chapitre 2. Même si, comme discuté par la suite, cette configuration est un peu différente de notre objectif, cela nous permet dans un premier temps de **vérifier que les textes générés sont utilisables**. Cela permet également d'introduire les différents problèmes de cette approche. Le scénario de substitution correspond aux cas où les données ne peuvent pas être distribuées, pour des raisons de confidentialité [10] ou de propriété intellectuelle. Nous apprenons tout d'abord des modèles de langue sur les données originales afin de générer des textes appartenant aux différentes classes du jeu de données. Nous étudions ensuite si ces textes générés permettent d'améliorer les performances lorsqu'ils sont ajoutés au jeu original et l'écart de performance lorsqu'uniquement ces données synthétiques sont utilisées. Nous étudions ces scénarios sur deux types de classifieurs : des modèles neuronaux et des modèles qui utilisent des représentations sac-de-mots. Les modèles sac-de-mots sont moins performants mais possèdent l'avantage d'être explicables, ce qui est particulièrement utile pour des modèles de détection de désinformation. Les résultats montrent qu'**un filtrage est nécessaire** afin de s'assurer que les données générées correspondent aux classes attendues. Une fois ce filtrage réalisé, les données générées permettent d'obtenir des **résultats similaires aux performances initiales lorsqu'elles sont utilisées en remplacement** du jeu de données original. Quand elles sont ajoutées **en supplément des données originales, elles permettent d'augmenter les performances, surtout pour les modèles sac-de-mots** qui bénéficient énormément de la diversité dans les formulations engendrées par la génération.

Cependant, lorsque l'une des classes est composée uniquement d'exemples synthétiques et l'autre uniquement d'exemples organiques, le classifieur peut exploiter les artefacts de génération et différences dans les distributions afin de prendre sa décision. De tels indices permettent au modèle d'atteindre une grande précision en apprenant des caractéristiques qui ne sont pas liées à la tâche cible et ne seront pas applicables en pratique sur des

échantillons organiques. Le modèle apprend donc à détecter les textes générés plutôt que la désinformation. Étant donné que les discriminateurs sont capables de détecter les textes générés, le paradigme des réseaux antagonistes génératifs (*Generative Adversarial Networks*, GAN) propose de les utiliser pour entraîner le générateur à produire des textes indétectables. Cependant, la nature discrète du texte rend l'apprentissage antagoniste complexe, puisque le gradient du discriminateur ne peut pas être rétro-propagé dans le générateur comme dans le cas des images. L'apprentissage par renforcement est donc souvent utilisé, en générant des exemples et en apprenant de ceux considérés comme étant les plus ressemblants à des textes humains par le discriminateur. Cependant, **l'espace des textes possibles (engendré par toutes les séquences de mots possibles) est si vaste qu'il est compliqué pour le générateur de trouver des exemples corrects obtenant de bonnes récompenses** (score du discriminateur) qui lui permettront d'apprendre.

Génération coopérative Pour surmonter ce problème, nous avons proposé d'utiliser la **génération coopérative**, c'est-à-dire d'utiliser le score du discriminateur directement pendant la génération afin de guider le modèle génératif vers les bons textes. Les méthodes coopératives existantes souffraient d'un manque de vision à long terme des textes en cours de génération. En effet, les modèles génératifs de textes sont dits auto-régressifs, c'est-à-dire qu'ils génèrent les mots les uns après les autres. Ainsi, une vision à long terme du processus de génération est un composant essentiel pour générer des textes cohérents et éviter d'aller vers des séquences qui ne peuvent pas obtenir de bons scores. Nous avons donc proposé l'utilisation de la **recherche arborescente Monte-Carlo (*Monte Carlo Tree Search*, MCTS)**, un algorithme qui sélectionne des actions à court terme en se basant sur des résultats à long terme, afin de guider la génération. Nous montrons dans un premier temps dans le Chapitre 5 que cette approche permet de générer des textes qui satisfont une contrainte définie par un classifieur externe, telle qu'un sentiment ou un sujet, grâce à sa vision à long terme.

Nous avons ensuite utilisé les textes générés coopérativement afin d'entraîner le générateur. Ainsi, dans le Chapitre 6, nous introduisons **une nouvelle formulation des GANs pour des données discrètes qui offrent des garanties de convergence théoriques** (qui n'existaient pas pour les approches précédentes). Ces dernières s'obtiennent en échantillonnant une distribution qui est une combinaison de la distribution du générateur et celle du discriminateur. Comme l'échantillonnage de cette distribution exacte est impossible

à cause de coûts trop élevés, nous introduisons des poids d'échantillonnage préférentiel. Cependant, il est toujours nécessaire d'échantillonner selon une distribution la plus proche possible de la distribution cible afin de limiter la variance. Nous utilisons donc la méthode de génération coopérative se basant sur le MCTS précédemment introduite afin d'échantillonner des textes qui obtiennent de bons scores à la fois pour le générateur et le discriminateur et se rapprocher de la distribution jointe. Les modèles entraînés suivant cette méthodologie, que nous appelons réseaux génératifs coopératifs (*Generative Cooperative Networks*, GCN) obtiennent des résultats à l'état de l'art dans différentes tâches de génération de langage naturel.

Nous avons également exploré l'utilisation de la génération coopérative pour **générer des textes représentatifs de la distribution effectivement apprise par un discriminateur afin d'améliorer son explicabilité** dans le Chapitre 7. Cela permet d'étudier son comportement et de trouver des biais potentiels sur le domaine défini par le générateur. En plus de fournir des explications globales, cette approche permet d'étudier le comportement d'un modèle sans nécessiter de données d'entrée, ce qui est particulièrement utile lorsqu'il n'y a pas de données représentatives disponibles. Aussi, la recherche de caractéristiques importantes pour le modèle est dirigée par le modèle génératif plutôt que par des exemples, ce qui permet une recherche plus large.

Enfin, nous avons étudié **l'impact du choix de l'architecture du discriminateur sur la qualité de son guidage ainsi que la complexité calculatoire du processus de génération coopérative** dans le Chapitre 8. Bien que les Transformers qui utilisent l'attention bidirectionnelle soient préférés pour les tâches discriminatives, ils ne sont pas adaptés à la génération coopérative, car ils nécessitent de recalculer les représentations des mots à chaque étape de génération. Nous montrons ainsi que **l'attention unidirectionnelle, qui permet de réutiliser les représentations précédemment calculées, n'entraîne qu'une faible baisse de précision**, qui n'a qu'un léger impact sur la qualité de la génération. Cependant, **cet écart est compensé par la différence en coût de calcul**, qui peut être réinvestie afin de combler l'écart de précision, tout en restant plus rapide que les discriminateurs bidirectionnels. Étonnamment, les discriminateurs génératifs, qui semblaient très intéressants à première vue en raison de leur capacité à évaluer tout le vocabulaire en une fois, permettant une recherche plus large, donnent de mauvais résultats. En effet, évaluer tout le vocabulaire en même temps se fait au prix d'une plus faible précision, qui se reflète à nouveau sur la génération. Le MCTS explorant plus en profondeur qu'en largeur, il ne bénéficie pas de la largeur offerte par les discriminateurs

génératifs tout en étant pénalisé par le signal de guidage de moins bonne qualité.

Perspectives ouvertes par la génération coopérative Bien que nous ayons étudié de manière approfondie la génération coopérative durant cette thèse, il reste encore **de nombreux cas d'utilisation à explorer**. L'application la plus directe consisterait à l'utiliser pour intégrer directement le filtrage du Chapitre 2 directement dans le processus de génération. Il serait également intéressant d'étudier les avantages de la réalisation de plusieurs boucles de génération/entraînement pour des tâches de classification, à la manière des GANs.

Aussi, un des composants principal du succès des modèles de langue actuel comme GPT-4 [249] et Llama 2 [347] est **l'apprentissage par renforcement à partir de retours humains (*Reinforcement Learning from Human Feedback*, RLHF)**. Dans cette configuration, le modèle de langue est entraîné afin d'optimiser une récompense définie au niveau de la séquence entière en utilisant l'apprentissage par renforcement, comme pour les GANs textuels. La différence réside dans la récompense : au lieu du score d'un discriminateur qui détecte les échantillons générés, le générateur optimise le score d'un modèle entraîné afin de modéliser les préférences humaines. L'entraînement du modèle de récompense se fait à l'aide de paires de séquences classées par des annotateurs humains, et le modèle a pour tâche de produire un score plus élevé pour la meilleure séquence de la paire. Le modèle est ensuite entraîné afin de produire des séquences qui maximisent cette récompense, et donc l'alignement avec les préférences humaines plutôt qu'être indiscernable des séquences organiques. Cela nécessite évidemment des annotations coûteuses qui ne sont pas nécessaires pour les GANs, mais cela semble être un moyen très efficace d'aligner les modèles génératifs avec nos besoins et nos valeurs, reflétés dans les annotations. Bien que le modèle de récompense soit fixé, empêchant la boucle d'apprentissage positive des GANs (et donc celle de GCN), **la génération coopérative pourrait être utile dans ce paradigme**. Comme avec le discriminateur des GANs, elle pourrait aider le générateur à produire de meilleures séquences et à obtenir des récompenses plus élevées en le guidant avec le modèle de récompense. Ainsi, ces meilleures séquences pourraient améliorer l'entraînement par renforcement ou être utilisées pour l'apprentissage par imitation, comme dans SelfGAN [304].

Des cas d'utilisations plus éloignés de nos études ont déjà été proposés, notamment la génération contrainte par la cohérence factuelle [400], les contraintes logiques [212] ainsi que les tests unitaires pour la génération de code [408]. La génération coopérative

pourrait également être utile dans le domaine des exemples adversaires, en utilisant le signal du discriminateur pour générer un exemple aussi proche que possible de la frontière de décision tout en étant mal classé [122]. L’exploration d’autres méthodes de génération coopérative qui tirent parti de l’exploration en largeur des discriminateurs génératifs est également une piste prometteuse. Pareillement, des approches de guidage sans classifieur ont récemment été proposées pour les modèles de langue [293]. On peut voir cette approche comme **un cas spécial de discriminateur génératif et elle peut donc être combinée avec les approches coopératives de la même manière**. Enfin, bien que nous nous soyons concentrés sur la génération de textes, nos études s’appliquent également à toute donnée discrète, telle que des données qualitatives [30].

Alignement cross-modal par la génération La génération coopérative permet de générer des échantillons qui trompent le discriminateur à un instant donné. Malheureusement, ce dernier s’adapte très rapidement et atteint à nouveau un taux de détection élevé. Cependant, même si les modèles génératifs ne permettent pas de créer une classe entièrement synthétique pour le jeu de données d’entraînement, leur amélioration reste bénéfique pour d’autres types d’augmentation de données qui utilisent de tels modèles, comme celle présentée dans le Chapitre 2. De plus, comme indiqué précédemment, les objectifs génératifs servent d’excellentes tâches proxys afin d’apprendre des bons espaces de représentations [405, 269, 28, 257, 348, 227, 397, 368, 332, 364]. Ainsi, afin d’intégrer les images dans cet environnement coopératif, nous avons exploré l’utilisation de modèles génératifs afin de créer un alignement cross-modal. Cet alignement est primordial afin de pouvoir comparer et mettre en relation les images associées aux textes des articles et ainsi étudier au mieux l’information qu’ils contiennent. **Afin de créer un alignement le plus fin possible**, nous nous concentrons sur la tâche de **génération de légende d’image distinctive** dans le Chapitre 10. Cette tâche consiste à générer une légende très descriptive de l’image d’entrée et de cette image uniquement. Plus spécifiquement, nous avons montré que les légendes humaines peuvent être utilisées durant un apprentissage par renforcement qui utilise des récompenses issues d’un modèle de recherche cross-modal. Ces légendes humaines permettent d’ancrer l’apprentissage dans la distribution humaine. Nous montrons qu’elles peuvent être **utilisées à la place des échantillons générés dans l’objectif d’apprentissage par renforcement**. Nous les utilisons également pour **entraîner un discriminateur à détecter les légendes générées des légendes humaines**, afin de régulariser l’apprentissage et s’assurer que les légendes générées ressemblent bien

à celles écrites par les humains. Cet ancrage est une première étape pour entraîner les deux modèles conjointement tout en évitant de dévier des textes humains. Enfin, nous introduisons **des récompenses contrastives, qui considèrent tous les éléments d'un batch comme des baselines pour la récompense**. Cela permet au générateur d'apprendre seulement des meilleures séquences. De plus, cette récompense contrastive permet de **considérer les deux directions de recherche cross-modal** (image vers texte et texte vers image), permettant de générer des légendes très descriptives de l'image d'entrée et de cette image uniquement. Étant donné que les modèles de génération de légendes sont de simples modèles de langue avec un conditionnement supplémentaire sur l'image d'entrée, des légendes obtenant de grands scores pour un modèle de recherche cross-modal peuvent être obtenues en utilisant PPL-MCTS pour guider le générateur avec le modèle de recherche. Ainsi, l'exploration de **la création d'une configuration similaire à GCN pour les données multimodales est une piste prometteuse**. Cela indique aussi que **Therapy s'applique également à des modèles multimodaux**, à condition qu'une des modalités soit le texte.

Les capacités et l'accessibilité grandissantes des modèles génératifs Les modèles génératifs ont bien évolué depuis ELIZA [378]. La qualité des textes générés augmente à un rythme incroyable et qui ne semble pas ralentir. Il n'y a pas si longtemps, la distribution publique du modèle GPT-2 avait été retardée car le modèle était considéré comme "trop dangereux pour être distribué"¹⁰. Bien qu'il n'existe toujours pas de consensus sur la question de la distribution publique [325, 405], cela peut faire sourire étant donné la capacité des modèles publiquement disponibles aujourd'hui. De nouveaux modèles toujours plus performants sont publiés tous les jours (FreeWilly2 [5] a été littéralement publié deux jours après Llama 2 [347]). Le phénomène est tel que des articles de revue sont nécessaires pour suivre le rythme [412]. Bien qu'il soit facile de se perdre dans ce nombre croissant de modèles, il n'a jamais été aussi facile de les utiliser, grâce à des bibliothèques telles que l'écosystème de [Hugging Face](#) qui permettent à tous d'accéder aux derniers modèles et jeux de données en quelques lignes de code. Même l'entraînement et l'inférence de tels modèles deviennent plus faciles avec des méthodes telles que LoRA [147] et la quantification [87, 86]. Ils peuvent même désormais s'exécuter sur du matériel non spécialisé^{11 12}.

10. <https://www.theverge.com/2019/11/7/20953040/openai-text-generation-ai-gpt-2-full-model-release-1-5b-parameters>

11. <https://github.com/ggerganov/ggml>

12. <https://github.com/huggingface/peft>

Les possibilités offertes par les modèles génératifs Ces modèles sont extrêmement puissants et peuvent déjà être utilisés comme des assistants pour une grande variété de tâches de la vie quotidienne. Il est d'ailleurs attendu qu'ils aient un impact considérable sur le marché du travail¹³ [94]. Ainsi, ils devraient faciliter le travail de vérification de l'information en prétraitant l'énorme quantité d'information disponible sur Internet et signalant les plus suspects. Ils pourront également fournir des indices aux humains chargés de valider la décision définitive. C'est particulièrement le cas pour les modèles génératifs augmentés par la recherche de données [130, 195, 42, 154], qui ancrent leur processus de génération dans des faits connus et vérifiés préalablement récupérés. Ces sources peuvent ensuite être transmises à l'opérateur humain afin d'accélérer sa vérification. Ces preuves externes s'avèrent particulièrement utiles pour lutter contre la réutilisation de contenu [2].

Aussi, comme mis en évidence dans cette thèse, les modèles génératifs peuvent être utilisés pour générer des données d'entraînement pour d'autres modèles. Récemment, ce processus a été utilisé en générant des légendes d'images pour remplacer le filtrage de légendes bruitées venant d'Internet [243] et l'entraînement d'un modèle de langue à l'aide d'exemples générés à partir d'un modèle de langue plus performant [340].

Enfin, en plus d'être un bon objectif pour apprendre des bonnes représentations des différentes modalités d'entrée, les tâches génératives permettent d'interroger le modèle sur l'espace de représentation appris, afin d'examiner ses connaissances. Par exemple, utiliser un modèle de diffusion pour générer une image "d'un fauteuil en forme d'avocat" permet d'obtenir des informations sur la manière dont ces concepts sont encodés dans l'espace de représentation.

Les risques des modèles génératifs Cependant, malgré les nombreux aspects positifs des modèles génératifs, ils possèdent également de nombreux risques. En effet, les humains sont facilement trompés par ces modèles et ne peuvent pas détecter de manière fiable les textes synthétiques [405, 71, 151], rendant le test de Turing [349] obsolète. Même s'il est possible d'améliorer notre capacité de détection avec des entraînements appropriés [92], une exposition permanente à des textes générés couplée à une augmentation des performances rendra la détection humaine de plus en plus difficile. Cela va augmenter les risques et l'efficacité d'une utilisation malveillante de ces modèles, notamment à des fins de campagne de désinformation automatique [119, 377]. Malheureusement, même avec un entraînement

13. <https://www.goldmansachs.com/intelligence/pages/generative-ai-could-raise-global-gdp-by-7-percent.html>

et des mesures de sécurité appropriées afin d’empêcher la génération de contenus nuisibles, les modèles peuvent être attaqués [429] et être utilisés à de mauvaises fins.

On pourrait penser que le fait que les discriminateurs soient capables de détecter les textes générés, bien que problématiques pour nos besoins, offre une solution à la désinformation générée par les réseaux de neurones [72, 25, 159]. Cependant, même sans atteindre l’objectif d’obtenir une distribution de textes générés indiscernable de celle des textes humains, la tâche n’est pas si simple. En effet, les discriminateurs sont fortement spécialisés sur le mode de génération du générateur [302, 151, 343, 25, 325, 350]. Cela signifie que tout changement dans le processus de génération, comme la température ou la méthode d’échantillonnage, fait s’effondrer les performances du discriminateur [16]. Cela est bien évidemment sans parler d’un modèle totalement différent, a fortiori lorsque ce dernier est entraîné sur un autre domaine. C’est un problème majeur, car en plus du nombre croissant de modèles de langue différents disponibles, un attaquant peut simplement ré-entraîner le modèle ou changer un peu le processus de génération afin de tromper le discriminateur. Discriminer entre la distribution des textes humains et plusieurs autres distributions très similaires est une tâche extrêmement complexe. Les bonnes performances des discriminateurs dans le contexte des GANs sont dues à une exposition suffisante à des échantillons générés par le modèle. Cette exposition est loin d’être garantie dans le monde réel, surtout pour les acteurs malveillants [56]. De plus, comme tout classifieur neuronal, les discriminateurs sont sensibles aux attaques adverses [253], qui permettent de créer des textes spécifiquement conçus pour exploiter les faiblesses du modèle et le tromper [122] afin de ne pas être détectés [383]. Enfin, de meilleurs discriminateurs peuvent être utilisés afin d’entraîner de meilleurs modèles génératifs et la génération coopérative présentée dans cette thèse permet de créer des textes qui sont, par construction, indétectables par le discriminateur utilisé comme guide.

En prenant en compte tous ces facteurs, même si nous arrivons à entraîner un discriminateur générique suffisamment performant, les taux de faux positifs resteraient trop élevés pour qu’il soit utilisé dans un contexte réel. Même avec un taux de détection presque parfait, lorsqu’il est appliqué à l’échelle réelle, de nombreux faux positifs vont apparaître. Si un faux positif n’est pas vraiment préjudiciable lors de l’entraînement d’un générateur, disqualifier des concurrents honnêtes d’un concours de photos¹⁴ ou pénaliser un étudiant qui n’a pas triché¹⁵ n’est pas acceptable. Pourtant, ce climat de méfiance est compréhensi-

14. <https://news.artnet.com/art-world/australian-photographer-disqualified-ai-generated-2337906>

15. <https://www.washingtonpost.com/technology/2023/05/18/texas-professor-threatened-fail-class-chatgpt-cheating/>

ble, étant donné que les modèles génératifs peuvent être et sont effectivement utilisés pour tricher aux examens [21]¹⁶ ou remporter des prix de photographie¹⁷. Cela est d'autant plus préoccupant que ces faux positifs peuvent être biaisés envers une communauté spécifique [203]. Sans mesures supplémentaires, la détection des textes générés semble trop peu fiable [291, 352], à tel point que même OpenAI a fermé son détecteur "en raison de son faible taux de précision"¹⁸ (26% de vrais positifs et 9% de faux positifs).

De plus, lors du début de la thèse, la capacité de génération des modèles de langue était encore limitée, sans parler de la capacité de générer des images. Seuls des réseaux très spécialisés, tels que les réseaux de génération de visages, étaient capables de produire des échantillons acceptables. Mis à part les chats, les chiens et les visages, les images générées étaient encore très éloignées d'images réelles. Ce n'était pas une option viable pour notre approche, car le contrôle sur les échantillons générés était inexistant. Cependant, moins d'un an après la sortie initiale de Dall-E [273], le premier modèle permettant la génération d'images convaincantes à partir d'un texte descriptif en *zero-shot*, les modèles de diffusion [281] s'améliorent de plus en plus, offrant un contrôle plus précis sur les images générées. Il est désormais possible de générer une image indiscernable d'une image réelle pour un être humain avec seulement une courte description de l'image désirée. La génération d'autres modalités, comme la vidéo et la voix, bien qu'un peu en retard, progressent rapidement et connaîtront certainement la même progression exponentielle et soulèveront également plusieurs cas d'utilisation problématiques [319, 141, 97, 192, 362], notamment l'usurpation d'identité. Des outils très convaincants sont déjà disponibles pour le grand public^{19 20}.

Atténuer les risques avec le tatouage numérique Le tatouage numérique peut faire partie de la solution, que ce soit pour les images [105, 360, 425] ou les textes [175, 1, 413, 68, 183], mais il doit être intégré **dans** le modèle et non au moment de la génération [105], de sorte qu'il soit coûteux pour l'attaquant de le supprimer²¹. Le tatouage numérique fonctionne en modifiant la distribution de sortie des générateurs pour ajouter une trace

16. <https://www.theguardian.com/technology/2023/may/18/ai-cheating-teaching-chatgpt-students-college-university>

17. <https://www.bbc.com/news/entertainment-arts-65296763>

18. <https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text>

19. <https://runwayml.com/>

20. <https://elevenlabs.io/>

21. <https://medium.com/@steinsfu/stable-diffusion-the-invisible-watermark-in-generated-images-2d68e2ab1241>

invisible à l'image ou au texte généré, facilitant sa détection ultérieure. Cependant, cette option nécessite encore de l'exploration, par exemple afin d'étudier la résistance aux attaques [176, 183, 291]. Dans le cadre de cette exploration, **notre récente étude [104] consolide les schémas de tatouage numérique existants** pour les grands modèles de langue génératif de trois manières différentes. Premièrement, nous introduisons de **nouveaux tests statistiques qui offrent des garanties théoriques robustes**, même à faibles taux de faux positifs, rendant la détection plus fiable. Deuxièmement, nous mesurons la **dégradation des performances induite par le tatouage sur tâches pratiques de traitement du langage naturel**, au lieu de mesurer les distorsions par rapport à la distribution originale. Cela donne une information plus précise sur l'utilité du modèle pour les tâches en aval, là où réside le véritable intérêt des modèles génératifs. Enfin, nous introduisons la **parallélisation de la détection de plusieurs tatouages numériques, transformant le tatouage binaire (présent ou non) en tatouage multibits (si tatoué, obtenir l'identifiant correspondant)**. Cela permet d'ajouter un message au texte tatoué, tel que l'utilisateur ayant généré le texte ou la version du modèle utilisée. Bien que le tatouage numérique soit une solution prometteuse pour la distribution publique de modèles génératifs et que les grandes entreprises d'IA semblent disposées à ajouter de tels tatouages numériques à leurs modèles génératifs^{22 23}, les acteurs malveillants peuvent toujours entraîner leurs propres modèles à partir de zéro, ré-entraîner un modèle pré-entraîné afin de supprimer le tatouage numérique ou créer manuellement de la désinformation. Même si le tatouage peut être utile pour détecter le contenu généré, il est illusoire de penser que tous les utilisateurs vont ajouter le tatouage à toutes les images générées. Ainsi, une solution potentiellement plus viable serait de renverser le paradigme de tatouage et de l'ajouter aux contenus authentiques, par exemple, en ajoutant le processus de tatouage directement dans les appareils photo. Ainsi, l'absence de marque indiquerait que le contenu est possiblement généré. Cela implique qu'un contenu ne devrait être considéré comme digne de confiance uniquement s'il a été signé par un éditeur reconnu. Cela est cohérent avec l'hypothèse selon laquelle Internet va prochainement être rempli de contenus générés et que nous devrions commencer à partir du principe que tout est généré, sauf s'il est démontré que cela ne l'est pas.

22. <https://www.theverge.com/2023/7/21/23802274/artificial-intelligence-meta-google-openai-white-house-security-safety>

23. <https://www.deepmind.com/blog/identifying-ai-generated-images-with-synthid>

Combattre la désinformation Lutter contre la désinformation sur les réseaux sociaux nécessite un changement d'objectif de ces derniers, passant de la maximisation de l'engagement des utilisateurs à la maximisation de la qualité de l'information, que ce soit délibérément ou par le biais de réglementations gouvernementales [188]. Cela peut prendre différentes formes, telles que le blocage des trajets de propagation en utilisant l'analyse de propagation et en ciblant les sources de fausses informations (utilisateurs influents/*bots*). Bloquer la désinformation avant qu'elle ne soit vue est crucial, car même après avoir détecté la désinformation, la corriger reste un vrai défi. En effet, le démenti ne se propagera pas autant ni aussi rapidement que la fausse information originale [359]. De plus, il est vraiment difficile de changer l'opinion de quelqu'un après qu'il ait été convaincu par de fausses informations [83]. Certaines solutions ont été proposées, telles que suggérer des recommandations éducatives et personnelles d'informations certifiées et réfutation de fausses informations ainsi que mettre davantage en avant les debunkers [358].

Les réseaux sociaux commencent à agir, par exemple X (anciennement Twitter) demande aux utilisateurs de lire l'article complet avant de le repartager²⁴ pour éviter de propager des titres trompeurs. Ils ont également ajouté des "notes de la communauté", où chaque utilisateur peut ajouter des informations à une publication, pour donner du contexte ou signaler des informations incorrectes²⁵. Bien que cela permette aux utilisateurs de participer à l'initiative de vérification de l'information, cela est encore loin d'être suffisant pour contenir le problème²⁶. Le plus important est de créer des outils qui donnent aux personnes vérifiant l'information le pouvoir de l'IA. Bien que ces outils ne soient pas parfaits et ne le seront peut-être jamais au point de pouvoir détecter automatiquement et de manière fiable la désinformation, ils peuvent sans aucun doute accélérer le processus de vérification et aider à prendre la décision finale. Par exemple, pour indexer de très grandes bases de données multimodales afin de trouver des indices pertinents, mettre en évidence les possibles affirmations problématiques, détecter des réseaux de propagation de désinformation, etc. En ce qui concerne la désinformation, l'intelligence artificielle peut être à double tranchant, il est donc important de **donner aux fact-checkers le pouvoir de l'IA afin de nous assurer de bénéficier du meilleur de la constante amélioration de l'IA**²⁷ et de ne pas simplement en subir le pire.

24. <https://www.theverge.com/2020/9/25/21455635/twitter-read-before-you-tweet-article-prompt-rolling-out-globally-soon>

25. <https://help.twitter.com/fr/using-twitter/community-notes>

26. <https://www.poynter.org/fact-checking/2023/why-twitters-community-notes-feature-mostly-fails-to-combat-misinformation/>

27. <https://time.com/6300942/ai-progress-charts/>

TABLE OF CONTENTS

Introduction	1
Multimodal Misinformation: A Modern Day Threat	1
Automatic Multimodal Misinformation Detection	3
Embeddings	7
On the Difficulty of Collecting a Suitable Dataset	10
Plan	14
Publications	15
I Generating Training Data	19
1 Transformers for Natural Language Generation	20
1.1 Transformers	20
1.1.1 Attention Layers	20
1.1.2 Encoder and Decoder	23
1.1.3 Pre-training	25
1.2 Textual Transformers	26
1.2.1 Pre-training Objectives	26
1.2.2 Architectures	28
1.2.3 Transfer Learning	29
1.3 Decoding Methods for Natural Language Generation	30
2 Generating Artificial Texts as Substitution or Complement of Training Data	32
2.1 Data Augmentation for Natural Language Processing	33
2.2 Generating Artificial Data	35
2.2.1 Fine-Tuning the Language Model	36

TABLE OF CONTENTS

2.2.2	Text Generation	36
2.3	Experiments	37
2.3.1	Experimental Setting	37
2.3.2	Neural Classification	40
2.3.3	Bag-Of-Words Classification	42
2.4	Conclusion	44
3	Textual Generative Adversarial Networks	47
3.1	Detecting Generated Data	47
3.2	Reinforcement Learning and Sparse Rewards	49
3.2.1	Exposure Bias	49
3.2.2	Reinforcement Learning	49
3.2.3	Generative Adversarial Networks	50
3.2.4	Language Generative Adversarial Networks Falling Short	51
II	Cooperative Generation	53
4	Introduction to Constrained Generation	54
4.1	Class-Conditional Language Models	55
4.2	Cooperative Approaches	55
5	PPL-MCTS: Constrained Textual Generation Through Discriminator-Guided MCTS Decoding	58
5.1	Introduction	59
5.2	PPL-MCTS Method	60
5.3	Experiments	64
5.3.1	Experimental Setting	64
5.3.2	Automatic Metrics Results	68
5.3.3	Examples of Generation	69
5.3.4	Hyperparameters Exploration	70
5.3.5	Human Evaluation	72
5.4	Conclusion	73
6	Generative Cooperative Networks for Natural Language Generation	76
6.1	Introduction	77

6.2	Generative Cooperative Networks	78
6.3	Cooperating for Natural Language Generation	81
6.3.1	Learning Algorithm	81
6.3.2	Efficient Sampling	83
6.4	Experiments	86
6.4.1	Experimental Setting	86
6.4.2	Results and Discussion	88
6.5	Conclusion	91
7	Therapy: Global Explanation of Textual Discriminative Models through Cooperative Generation	93
7.1	Introduction	94
7.2	Existing Explanation Methods	95
7.2.1	Example-Based Explanations	96
7.2.2	Feature-Attribution Explanations	96
7.3	Therapy	97
7.4	Experiments	99
7.4.1	Experimental Setting	99
7.4.2	Correlation of the Explanations and the Glass-Box Weights	102
7.4.3	Precision/Recall of the Returned Features	104
7.4.4	Insertion/Deletion of Important Features	105
7.5	Conclusion	106
8	Which Discriminator for Cooperative Text Generation?	108
8.1	Introduction	109
8.2	Choosing the Right Teammate	109
8.3	Empirical Study	111
8.3.1	Discrimination Strength	112
8.3.2	Generation Quality	113
8.3.3	Computational Gain	114
8.4	Conclusion	116
III	Cross-Modal Generation	117
9	Cross-Modal Generation as an Alignment Task	118

TABLE OF CONTENTS

9.1	Multimodal Transformers	119
9.1.1	Cross-Modal Attention	119
9.1.2	Pre-training	120
9.2	Dual Encoder	121
9.2.1	Fast Retrieval	121
9.2.2	Contrastive Learning	122
9.3	Fast Dual Encoder and Slow Cross-Encoder	124
10	Distinctive Image Captioning: Leveraging Ground Truth Captions in CLIP Guided Reinforcement Learning	127
10.1	Introduction	128
10.2	Image Captioning	130
10.3	Method	133
10.3.1	Preventing Reward Hacking Using a Discriminator	133
10.3.2	Beyond a Single Baseline: The Bidirectional Contrastive Reward . .	134
10.3.3	Reward-Weighted Teacher Forcing	135
10.4	Experiments	136
10.4.1	Experimental Setting	136
10.4.2	Model Variants	137
10.4.3	Results	137
10.5	Conclusion	139
11	Conclusion and Perspectives	141
11.1	Overcoming the Training Data Collection Challenge Through Data Generation	141
11.2	Generative Models and Misinformation: A Double-Edged Sword	145
IV	Appendices	153
A	PPL-MCTS: Constrained Textual Generation Through Discriminator-Guided MCTS Decoding	154
A.1	Complementary Results	154
B	Generative Cooperative Networks for Natural Language Generation	159
B.1	Proof for Theorem 6.2.1	159
B.2	Proof for Theorem 6.2.2	161

B.3 Results with Former MLE Baseline	163
C Therapy: Global Explanation of Textual Discriminative Models through Cooperative Generation	164
C.1 Qualitative Results	164
Bibliography	169

INTRODUCTION

Multimodal Misinformation: A Modern Day Threat

The development of online social media has significantly increased people's ability to share opinions and emotions, upholding freedom of speech to a whole new level. As a result of the growth of social networks, Internet users rely heavily on these platforms as sources of information instead of relying on traditional news channels¹. This is particularly true for young people as shown in [222]. However, the quality of news on these networks is far lower and noisier than traditional media. Indeed, when it comes to information, social networks have two negative aspects that lead to the spread of misinformation, that is, misleading and erroneous information.

First, every user is a content creator; anyone can post anything, regardless of their expertise or honest intent and without having to be held responsible for these publications. Thus, false information appears (often referred to as fake news), which may be widely disseminated and go viral. The spread of such information can be due to a desire to manipulate (for political, commercial, or other reasons), or simply a lack of knowledge about the topic. Certain events have really catalyzed this phenomenon, such as the US presidential election of 2016 [7] and the COVID-19 pandemic [6], that was even considered as an **infodemic**^{2 3}, emphasizing how the misinformation spread from person to person, just as the virus itself. The negative aspects of such erroneous information are numerous, ranging from threats to democracy by political manipulations affecting elections [43, 423, 106] to significant impact on the monetary system⁴, notably through manipulation of the stock market⁵. Even more critically, it also leads to death of people, for example during

-
1. <https://www.pewresearch.org/journalism/2018/09/10/news-use-across-social-media-platforms-2018/>
 2. <https://www.washingtonpost.com/archive/opinions/2003/05/11/when-the-buzz-bites-back/bc8cd84f-cab6-4648-bf58-0277261af6cd/>
 3. <https://www.who.int/health-topics/infodemic>
 4. <https://www.zdnet.com/article/online-fake-news-costing-us-78-billion-globally-each-year/>
 5. <https://www.forbes.com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/>

incidents caused by the spread of misinformation (e.g., Pizzagate⁶ and India's mobs⁷) or through the spread of false information about health⁸ causing health policymakers defiance [124, 153].

Second, the activity and amount of information circulating on social networks are gigantic⁹, thus analyzing everything is particularly complex and costly. In addition, there is the remarkable speed at which information that has gone viral spreads [359].

This implies that social networks are filled with misinformation, and unfortunately, humans suffer from different biases that make them vulnerable to such information on such platforms. For example, multiple exposures to the same (false) information increase the trust in it because of the *validity effect* [39]. This adds to the *confirmation bias* [245] which makes humans more prone to trust something that confirms their initial beliefs or something that please them (*desirability bias* [108]). Finally, social aspects and the peer pressure can change our perception and behavior (*bandwagon effect* [193]).

For all these reasons, it is vital to detect fake news and mark it as such as quickly as possible because humans are bad at detecting deception [285, 420] and struggle to correct their beliefs once fake news has already gained their trusts [83, 31]. At best, misinformation should be flagged before entering the echo chambers [70] where it will be amplified and reinforced. Even though large human fact-checking are developped^{10 11 12 13}, manually checking each piece of information is far too lengthy and costly. Hence, automatic tools that can flag erroneous content and help human fact-checker to assert the veracity of a post are needed, especially for large social networks such as X (formerly Twitter) or Facebook.

Although most of the information is transmitted as text, on the Internet, it is often coupled to other modalities such as images or videos, to make the post more appealing, attract more attention [102] and engagement¹⁴, spread further than text alone [404] and increase the belief of the reader in the statements in numerous ways [131, 103, 328, 199, 8, 404, 138]. In addition to generating traffic and belief, which cause a larger dissemination of

6. <https://time.com/4590255/pizzagate-fake-news-what-to-know/>

7. <https://asiatimes.com/2019/10/indias-fake-news-problem-is-killing-real-people/>

8. <https://www.who.int/news-room/feature-stories/detail/fighting-misinformation-in-the-time-of-covid-19-one-click-at-a-time>

9. <https://datareportal.com/social-media-users>

10. <https://www.politifact.com/>

11. <https://www.poynter.org/news/fact-checking/>

12. <https://www.lemonde.fr/verification/>

13. <https://defacto-observatoire.fr/>

14. <https://www.fastcompany.com/3022116/whattwitters-expanded-images-mean-for-clicks-retweets-andfavorites/>

false information, the image part is also a vector of misinformation propagation. The most obvious case is the manipulation of the image content [129, 136, 93], especially with the development of powerful editing tools such as deepfakes [4, 232]. Yet the most widespread case is image repurposing¹⁵. Typically, it consists of re-using a previously posted image without altering it but modifying its description to convey false information. For example, in Figure 1, a picture taken after a festival representing a park littered with garbage has been repurposed to make the reader think that an environmental manifestation is at the origin of this pollution to discredit the movement. Similarly, when hurricanes occur, image repurposing is commonplace as users upload pictures of other hurricanes, often making the situation look worse than it is and causing panic. This is an extremely popular case of disinformation because it is really cheap and easy to produce compared to image manipulation. The attacker can write false information on any topic and simply use a picture that seems to support his claims to benefit from previously mentioned biases that come from adding an image to information. Besides, since the picture is legitimate, there is obviously no trace of manipulation [62] that could make the reader suspicious or give hints for the detection.

Please note that there are various types of false information¹⁶. For example, false information spread on purpose is referred to as *disinformation* [182] and, when unintentional, as *misinformation* [75, 152]. In this thesis, we are not interested in discriminating the intent [424]. We also do not consider *harmfulness* [374] in the detection even though harmful content can be correlated to false content [6]. Our interest lies in detecting a mismatch between the semantics of the image and the ones of the associated text [157, 290, 215, 156, 165, 13] and to assert the *factuality* [152] of the statements (w.r.t known facts) and connection of the visual and textual content.

Automatic Multimodal Misinformation Detection

Given the ever-growing threat of misinformation, many research efforts have been directed toward their automatic detection. The goal is to classify input news as true or false. The news can be composed of textual and visual content, as well as meta information about the news such as its propagation path in the network, the profile of the user posting

15. <https://theconversation.com/out-of-context-photos-are-a-powerful-low-tech-form-of-misinformation-129959/>

16. <https://firstdraftnews.org/articles/fake-news-complicated/>

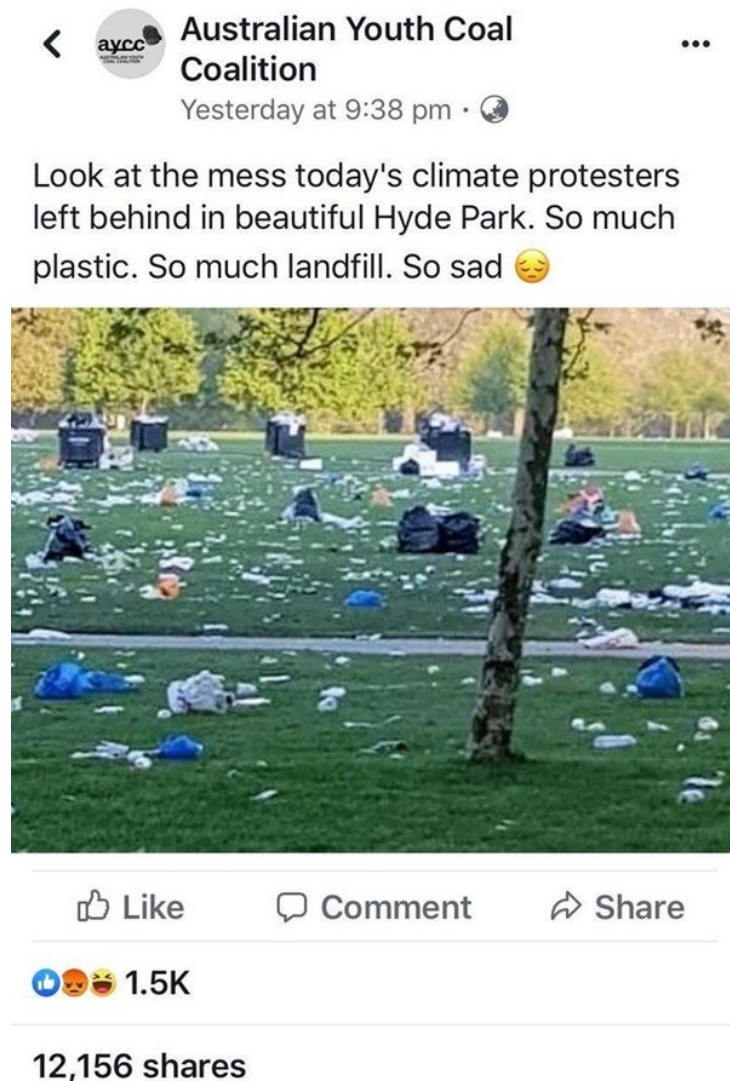


Figure 1 – Example of image repurposing. The original picture has been taken after the 420 festival, yet the [author claims](#) that it has been taken after an environmental demonstration in order to discredit the movement.

it, etc. Numerous approaches have been explored, using various types of clues that can help in the detection.

A first line of work uses metadata surrounding the content itself to detect misinformation based on its context. For example, the way news propagates can be helpful [415, 37, 313, 187], given that fake news tends to propagate faster and further than real news [359]. Another source of detection clues is social information about the poster and people sharing the information [55, 314, 287, 357, 322, 26] to evaluate the credibility of the source (writer and spreader). Notably, it allows to detect malicious users such as millions of bots that spread fake news [107, 351, 51, 235] and increase the activity of unreliable news in the early-stage to make it viral [309]. Detecting the main accounts spreading false information helps to fight against massive disinformation campaigns. At the intersection of these two kinds of approaches is network analysis [421, 314, 287], which studies the users' networks that emerge on online platforms and the way the information propagates through them. However, these approaches are not always applicable, for example for early stage detection, when the information is not shared on social networks or when there is no additional metadata about users or propagation available. This is damaging because it is the most worthwhile to take action in the early stages of propagation, before many users have been exposed to it.

Then, some methods try to detect false information by analyzing the content of the news itself, through the extraction of statistical or semantic features. Relying on the observation that deceptive texts are written differently [9, 422], some methods use linguistic and stylistic characteristics that identify false content. For example, the frequency statistics on lexicons and part-of-speech [422], the use of emoticons and symbols [55], or the language style [265, 223, 101, 259]. Similar approaches exist for images, which study the statistical differences between images used in real and fake news [164]. In addition, some methods derived from forensic features [36, 120, 220, 414] have attempted to detect image manipulations [41, 32, 416], generation artifacts [109, 200, 201, 225] or visual artifacts created by multiple compressions induced by successive resharing [52].

Other content-based methods study the semantics of the content [372, 419, 320, 64, 55, 173]. These approaches are the closest to the traditional definition of fact-checking by seeking to assert or refute the claims made in the news. This is the type of approach that we chose to study in this thesis. More specifically, we are interested in building representations of the text and associated image that allow to verify their semantics and the semantic consistency between the two modalities. Some approaches only leverage the

textual modality by detecting, extracting and verifying claims using retrieved evidence [128]. Such monomodal approaches discard the information coming from the image modality, and most importantly, the interactions between the two modalities. Early multimodal studies used the other modality as a simple complement, by concatenating the features of both modalities [372, 320, 321]. However, such light fusions of the two modalities fail to model the complex cross-modal interactions, which allow not only to get the detection power of both modalities but also to catch cases that are not detectable with only one modality, such as image repurposing. Thus, some studies have examined the consistency between an image and the associated text. [419] for example compares the caption generated for the input image to the associated caption by measuring their similarity, whereas [390] directly compare the two modalities in a common representation space. [118] extends this idea by using best-performing Transformers to obtain better representations. Besides the alignment of the two modalities, considering both modalities at the same time enables to detect what parts of the news are important features because they are present in both. This allows the monomodal representation to be refined by enhancing it using information from the other representation.

While directly asserting the semantic consistency allows to detect randomly shuffled pairs of image-caption [157], it is not sufficient to detect intelligent manipulations like the coherent replacement of a type of entity (e.g., the replacement of a personality by another personality or one place by another one) [290], swapping the image with a semantically close one [215], or generating a caption based on the ones of similar images [156, 165]. Hence, knowledge-based approaches use external sources to verify the knowledge in the content and fact-check it by asserting its consistency with known facts [156, 290]. This enables the detection of misinformation cases that do not have any visible flaws and require external knowledge to be asserted. Knowledge bases often take the form of knowledge graphs [244], built using a collection of triples (Subject, Predicate, Object) (SPO), e.g., "Emmanuel Macron is the president of France" can be represented as (EmmanuelMacron, Profession, President). The knowledge base is composed of verified SPO that are considered true. Fact-checking then consists of comparing the SPO of the knowledge base with those extracted from the claims of the news, for example, verifying that they are compatible and inferring the probability that a triple exists [69, 311]. Note that both constructing the knowledge base and extracting the claims requires extracting facts [244] from raw data sources (e.g., Wikipedia) using various methodologies (also based on deep networks) [89, 205, 396]. In addition, the main focus has been to build large knowledge bases rather than

update them quickly, which is necessary for early detection. We thus preferred to focus on comparing the information contained in raw data directly available in newspapers, without additionally casting it into graphs.

Content-based approaches require building a representation space where the modalities can be scrutinized and compared to other elements, from the same modality or the other one. These representations, also called embeddings, are also leveraged to retrieve useful information in knowledge-based approaches. Representation learning is thus a key component of such approaches, and the quality of the representation spaces is a key factor in detection performance.

Embeddings

Neural networks are built to work with vectors, therefore it is necessary to represent the input data in a way that is compatible with the expected input format. While each input can naively be represented by the binary representation of a unique identifier, this representation is not suitable because of its high dimensionality and lack of semantic information. The curse of dimensionality and lack of semantics lead to an absence of the notion of proximity between elements. The goal of **embeddings**, also called distributed representations [140] is to create a low-dimensional representation space where different concepts are encoded along its different axes. This space allows to represent the input data in a continuous space where elements that share some semantic similarities will be close to each other and unrelated ones will be far apart, as illustrated in Figure 2. Besides being in the form of continuous vectors, these embeddings should correctly describe the object (such as the meaning of a word) and have relevant properties, like a small distance between two vectors describing two similar words (in their spelling or meaning) and a large one if they are not similar at all.

While in the past, these vectors were encoding relevant features handcrafted by experts in the field, the current trend is the automatic extraction of features through deep learning networks trained on a given task, alleviating the feature engineering problem. Such techniques allow the learning of dense representations, i.e., embeddings, with multiple levels of representation obtained by the composition of the representation of the prior layers, starting from the raw input. Each layer then extracts a higher, more abstract representation, culminating in the last layer in a task-related representation (e.g., if a dog or a cat is pictured in the image). Although the embedding space should be as general

as possible, the proximity is directly tied to how it was built, as illustrated in Figure 2 with an embedding space designed to differentiate pictures of cats and dogs. Projecting the input data in this space creates a description of the elements that contains important discriminative features relevant to the downstream task. Semantic continuity creates a neighborhood of similar elements that enables clustering, retrieving similar elements, and generalizing the processing of the data to similar (unseen) elements.

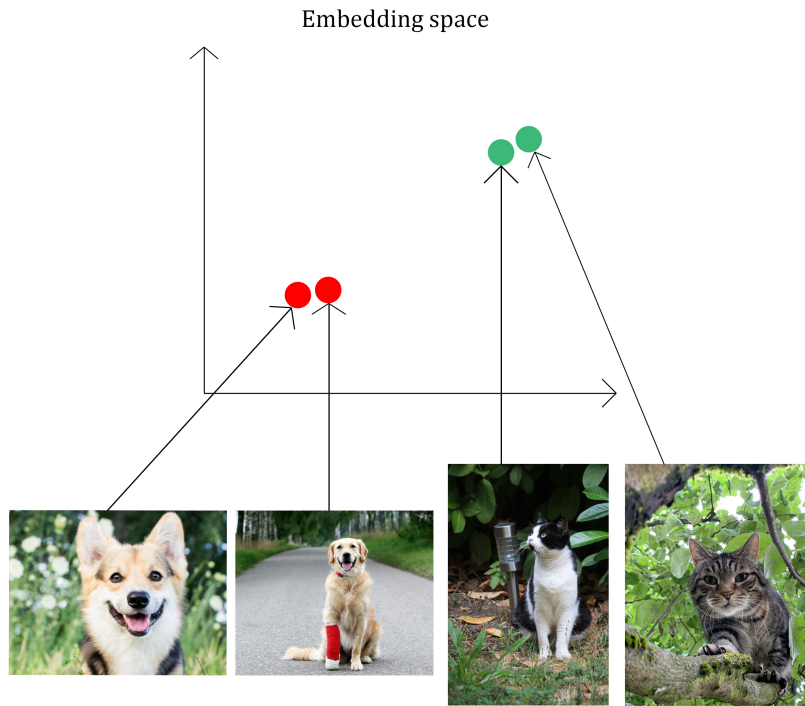


Figure 2 – Representation of an embedding space adapted to cats and dogs picture classification.

Image embeddings Before the Transformers [353], image embeddings were constructed using **Convolutional Neural Networks** (CNN) [191]. CNNs leverage convolutions, that are filters applied on local patches of the image as a sliding window and are activated when a discriminative pattern is found in an image. The filters are learned during the training to detect patterns that are useful for the training tasks, for example, image classification. These patterns are then combined through a succession of hidden layers to detect local conjunctions of features from the previous layer and increase the receptive field. CNNs became state-of-the-art in many computer vision tasks after VGG [318] made a huge improvement in the classification task. For a long time, the creation of more accurate

CNN-based models, and therefore better embeddings, involved stacking an increasing number of hidden layers [335], allowing the networks to create complex patterns that are highly discriminatory. However, this approach has been limited due to the vanishing gradient problem that arises when training very deep networks. The more layers the gradient goes through, the lower the gradient is, causing early layers to be barely updated. Skip-connection, that is, adding the output of the previous layer to the output of the following layer solves this issue by allowing the gradient to directly backpropagate through these identity layers, allowing earlier layers to be updated, even in a deep network. The ResNet model [134] is built using skip-connections and has improved state-of-the-art results leading to really precise embeddings. It has then been further enhanced with wider layers as in Wide ResNet [402] and an additional dimension beside width and depth (cardinality) in ResNeXt [388].

Text embeddings Similar to images, a lower-dimensional and more meaningful representation of texts can be found in the representation extracted by a network trained for a specific task. In one of the most well-known embedding techniques, **Word2Vec** [229, 228], the network is trained to predict words next to the input word in a corpus. Given that words with similar meanings are often found in the same context, this offers a good representation of the meaning of the word. The network is expected to return the one-hot encoded word, given words contained in a window around it as input. This objective is called Continuous Bag-of-Words (CBOW). The opposite task is called Skip-gram and predicts words contained in a window given the central word. The weights of hidden layers can then be used as continuous vectors encoding features related to the semantics of the input word.

While it can be useful to encode a single word, the text is often larger than a single word and can have various lengths, from a sentence to a whole document. Taking the average representations of every word in the sequence does not consider the interaction between them. **Recurrent Neural Network** (RNN) [288] considers text as a sequence of words. To do so, RNN works in the same way as a traditional neural network, with the addition of a state value as an input. This value represents information about the past elements of a sequence. This state is updated after processing each word in the text and used as an additional input to keep an abstract representation of the previous content. One can see RNN as a chain of neural networks sharing their weights, which takes, in addition to the next word, the value of the hidden state of the previous network as input. However, RNN

does not work well for very large inputs because it only accumulates information into a fixed-size vector and never forgets it, compressing an increasing amount of information in an already saturated vector. To correct this, **Long Short-Term Memory** (LSTM) [142] provides RNN with an explicit memory that mimics the natural behavior of the memory. This memory cell accumulates information by copying its state value and adding part of the information from the current input to it. This value is then multiplied by the value of another unit, the forget gate, which has a value contained between zero (to forget everything) and one (to keep everything) to control the longevity of the memory. The behavior of this unit is learned during training. Thus, the network can learn how to manage the content of its memory by adding only part of the information contained in the input and clearing part of it when needed, leading to significantly more efficient modeling of long-term dependencies.

Transformers The advent of the Transformers brought the quality of these representation spaces to a whole new level thanks to their large capacity brought by attention layers and heavy pre-training on very large scale unlabeled dataset using self-supervised objectives. This allows to learn a general representation space that can be used for a wide range of downstream tasks and achieves very strong results out of the box [88, 209, 135, 271, 91]. The BERT [88] model made many other methods obsolete because of how easy it was to achieve better results by just applying it to a given task with little to no fine-tuning. Following this success, ever better performing models have been proposed and extended to other modality as well as multimodal inputs. These incredible results led the community to focus on this architecture that became dominant. We will therefore focus on this architecture in the rest of this thesis. Details about the Transformers architecture and their pre-training are given in the beginning of Part I. Their application to multimodal inputs is discussed in Part III.

On the Difficulty of Collecting a Suitable Dataset

Even when using a pre-trained model, fine-tuning it on additional task-specific data usually yields large performance gains. Hence, it is required to adapt the model using a dataset that is representative of the task at hand to get the best performance possible. In the case of misinformation detection, the model should be trained on a dataset that contains both genuine and fake news. However, the creation of such a dataset is challenging

for several reasons. We consider three different approaches to create it: collecting the data, exchanging one modality to create a misalignment or generating the data.

Although one might think that the increase in misinformation might help in the collection of training data, it is difficult to create a suitable dataset that allows the training of models that effectively detect inaccurate information. Indeed, deep neural networks require huge amounts of data to function, and there is no "reliable source" of false information (satire aside [286]). Unlike the collection of genuine information that can be created using traditional channels, fake news must be collected manually after a fact-checking step. This significantly limits the amount of data that can be used to train the model. Although early studies focused on text-only data [369, 336, 296], a significant amount of datasets composed of images alongside with texts have been proposed [315, 238, 165, 40, 163, 430].

Yet, the difference in sources between the two collections results in biases. Since the collected genuine information is written by professional journalists, it consists of well-written and documented texts, while fake news collection is much more diverse and often suffers from easily observed linguistic and stylistic biases discussed in the previous section [422, 55, 265, 223, 101, 259]. Similarly, images can suffer from statistical biases and artefacts previously mentioned [41, 200, 201, 225, 52]. Given such differences in style and quality, these datasets can be solved by asserting the source rather than checking that the information is correct. Monomodal methods thus often easily achieve above random performances [148], emphasizing that multimodality is not strictly required to solve these datasets. This leads to biased models that learn non-generalizable features related to the training data (e.g., source identification), resulting in highly variable performances on different datasets [44]. The capacity of Transformers makes them very prone to fitting the noise contained in the data and learning detection features that will not necessarily hold on real-world data or evolve over time. This is especially true for multimodal inputs where the model will heavily rely on the biased modality [110, 292, 50] and not learn multimodal representations. Although some of these features may be of practical interest for detection, they do not reflect the veracity of the information and may not generalize to general data, either poorly reported true information or well-forged misinformation. Such biases are also very specific to the data sources and may not hold for different ones or other languages. Note that most methods relying on latent representations actually capture more of this kind of surface information than knowledge about our world, especially when external databases are not leveraged during inference. This is in contrast with the objective of

asserting the objective facts contained in news.

Finally, even if a sufficiently large and unbiased dataset was created, it would be out of date by the time it was created due to the dynamic nature of the information. Facts can change (e.g., the current President of the United States) and our knowledge can evolve. Something can be considered true one day and false the next, such as the geocentrism. Keeping up with traditional media can already be challenging, but the fact-checking stage takes too long to keep up with new conspiracies and rumors.

Another viable option is to forge misaligned image and article pairs by exchanging the original image of the article with another image [215, 157]. NewsCLIPpings [215] leverage recent breakthroughs in cross-modal retrieval of the CLIP model [270] to perform the swapping and yield very convincing and challenging fake couples. However, we chose to focus on generating synthetic articles for different reasons. First, this approach requires a somewhat related and illustrative picture to be associated with the text, which might not exist or be in the available database. More importantly, this approach relies on the quality of an existing cross-modal alignment to select a good replacement. On the other hand, learning to generate synthetic articles creates useful representations space that can be used for discriminative tasks, either for text inputs [405, 269], image inputs [28, 257] or multimodal ones [348, 227, 397, 368, 332, 364]. Thus, instead of leveraging the capacity of an already existing and fixed model, we learn a good representation space that can serve as a starting point to train the discriminative model on the downstream generated data or to retrieve data from a knowledge base.

In this thesis, we explore whether traditional media suffices to constitute the sole source necessary for **creating a training dataset for misinformation detection through the utilization of generative models**. These models are trained to mimic the distribution of their training samples, enabling the generation of data belonging to that distribution. In this way, the previously mentioned problems are solved, as it becomes possible to produce an arbitrary number of documents on baseless topics closely resembling authentic news articles. This effectively addresses the issues related to both quantity and domain bias. The model can be used to generate data on new topics as needed and eventually trained on new data to add new topics. Grover [405] demonstrates the training of a language model to write an article that follows the format and writing style of a given newspaper about a headline given as input. Then, modified versions of the original headlines can be given as input of such a model to create a fake article linked to the image of the original article [339]. To keep the article close enough to the image so that the fake is not too

obvious, the perturbation in the headline must be subtle. The replacement of an entity by another entity of the same type, such as a person or country, is a good example of a manipulation that can be misleading if the generated article provides information related to the new entity. Named-entity recognition models can be used to find such entities in the headline and replace them with another entity of the same type. An example of a fake article generated following this procedure, with Grover as the generator, is shown in Figure 3. While this yields convincing fake articles that easily fool humans [405], the generated texts do not follow the exact same distribution of the original articles, which gives obvious detection clues for a machine learning model.

The first part of this thesis introduces the use of generative models to create training data and its challenges. Chapter 2 shows that the generated data is noisy and requires filtering, but that this does not totally prevent the use of generated data to train discriminative models. However, the next chapter introduces that easily detecting the source of the data hurts the training when the classifier can rely on this information to do the classification. This motivates our explorations on methods to enhance the generative models to close the gap between the two distributions, notably through the generative adversarial networks paradigm.

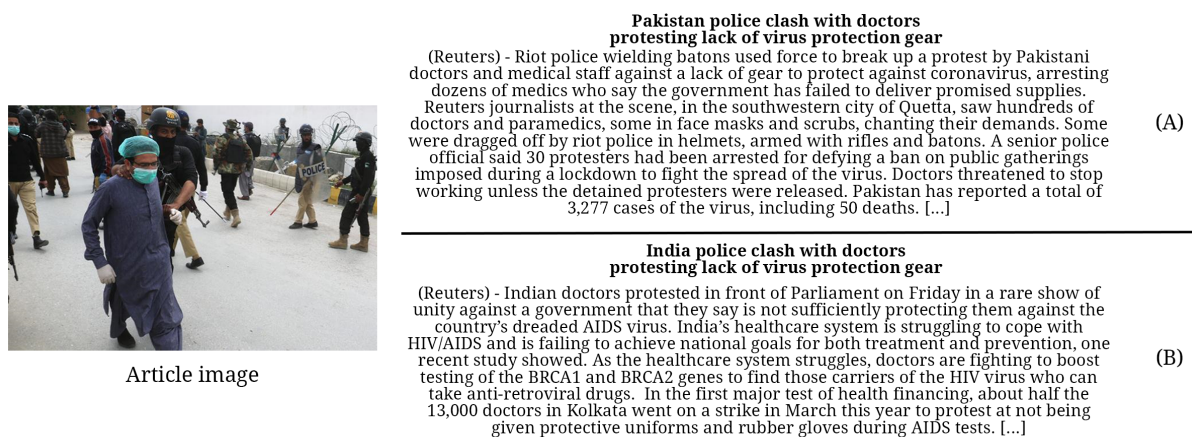


Figure 3 – Example of an original article from Reuters (A) and one generated following our methodology with Grover [405] (B), headlines are in bold.

Plan

In this thesis, we explore the use of synthetic, generated, textual data to train discriminative models. Please note that, in this manuscript, we have deliberately kept the original title of the thesis offer, even if, as we shall see, some of our contributions stray from the initial applicative goal of misinformation detection. However, we felt it important, in this introduction and in each of the chapters, to place these contributions in the context of the objective of analyzing and detecting misinformation, which is at the root of our motivation for this thesis work.

In Part I, we introduce the Transformer architecture and its application to natural language processing tasks, notably Natural Language Generation (NLG). We then show that generated texts can be used as a complement or in replacement of the original training dataset for textual classification tasks in Chapter 2. Chapter 3 presents the issue that arises when one class is composed of synthetic samples and the other of organic ones: it is easier for the classifier to become a discriminator able to distinguish real from generated samples. It then introduces how Generative Adversarial Networks (GANs) leverage said discriminators to train and improve generative models, and the challenge of applying this paradigm to discrete data such as texts.

Part II introduces cooperative generation, where an external model (e.g., a discriminator) is used to guide the generation process. After an introduction to the task of constrained generation in Chapter 4, Chapter 5 presents our novel cooperative generation method that leverages the Monte Carlo Tree Search to guide the generation and its performance on generating text satisfying a constraint defined by an external classifier. Chapter 6 then introduces Generative Cooperative Networks, a framework that uses a discriminator detecting generated texts in cooperative generation and trains the generative model using cooperatively generated texts. Chapter 7 presents a novel application of cooperative generation to the explainability of textual discriminative models. Finally, Chapter 8 explores the impact of the choice of the guiding model on the guidance quality and the computational complexity of cooperative generation.

Part III links previous parts to the multimodal aspect of this thesis by the mean of the image captioning objective to create cross-modal alignment. We first introduce multimodal Transformers and cross-modal attention layers. We then introduce the dual encoder architecture trained using contrastive loss and discuss the pros and cons of the different architectures for cross-modal retrieval. Chapter 10 introduces our preliminary

work on Multimodal Generative Cooperative Networks, which explores how ground truth captions can be leveraged in a reinforcement learning training using rewards from a cross-modal retrieval model to ground the training to the original distribution.

Finally, Chapter 11 concludes this thesis by summarizing the contributions and introducing their perspectives in Section 11.1. The usefulness of generative models for misinformation detection but also the risk of large-scale automatic dissemination brought by ever-improving generators are then discussed in Section 11.2. This section also briefly introduces watermarking as a potential solution to these risks and presents our contribution to consolidating this avenue.

Publications

This manuscript is based on different studies that have been published at national and international conferences or are under review to be published. Please note that, since these works cover a wide range of topics, this manuscript does not contain a proper "state-of-the-art section" as is usually the case. The information relevant to understanding and positioning each study is given throughout the manuscript.

Although I am not the first author of all these publications, I made substantial contributions to every study presented in this manuscript. For the sake of transparency, details about those contributions will be given in the introduction of chapters where I am not the first author.

International Publications First, we introduce the different publications at international conferences:

- Vincent Claveau, Antoine Chaffin, Ewa Kijak. "Generating artificial texts as substitution or complement of training data". Proceedings of the 2022 Language Resources and Evaluation Conference (LREC). 2022
Explore the use of artificially generated texts to train models, either as an addition to original training data or as a total substitution. It is introduced in Chapter 2.
- Antoine Chaffin, Vincent Claveau, Ewa Kijak. "PPL-MCTS: Constrained Textual Generation Through Discriminator-Guided MCTS Decoding". Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). 2022

Introduce a cooperative decoding strategy that leverages the Monte Carlo Tree Search to guide a generative language model using an external model. It is introduced in Chapter 5.

- Sylvain Lamprier, Thomas Scialom, Antoine Chaffin, Vincent Claveau, Ewa Kijak, Jacopo Staiano, Benjamin Piwowarski. "Generative Cooperative Networks for Natural Language Generation". Proceedings of the 39th International Conference on Machine Learning (ICML). 2022

Add the cooperative aspect to traditional textual Generative Adversarial Networks trained using reinforcement learning. It is introduced in Chapter 6.

- Antoine Chaffin, Thomas Scialom, Sylvain Lamprier, Jacopo Staiano, Benjamin Piwowarski, Ewa Kijak, Vincent Claveau. "Which Discriminator for Cooperative Text Generation?". Proceedings of the 45th International Conference on Research and Development in Information Retrieval (SIGIR). 2022

Empirically measure the complexity/quality trade-off of different types of Transformer-based discriminative models when used in cooperative generation approaches. It is introduced in Chapter 8.

- Antoine Chaffin, Julien Delaunay. "“Honey, Tell Me What’s Wrong”, Global Explainability of Textual Discriminative Models through Cooperative Generation". Proceedings of the Sixth Workshop on Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP). 2023

Leverage cooperative generation to generate global explanations of any textual discriminative model without requiring input data. It is introduced in Chapter 7.

- Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, Teddy Furon. "Three Bricks to Consolidate Watermarks for Large Language Models." Proceeding of the 2023 International Workshop on Information Forensics and Security (WIFS). 2023

Consolidates existing methods to watermark the texts generated by generative language models. It is discussed in the perspectives of this thesis in Section 11.2.

National Publications Some of our studies have also been redacted in French to be introduced to the national community:

- Vincent Claveau, Antoine Chaffin, Ewa Kijak. "La génération de textes artificiels en substitution ou en complément de données d’apprentissage". Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles (TALN). 2021

- Antoine Chaffin, Vincent Claveau, Ewa Kijak. "Décodage guidé par un discriminateur avec le Monte Carlo Tree Search pour la génération de texte contrainte". Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles (TALN). 2022
- Antoine Chaffin, Thomas Scialom, Sylvain Lamprier, Jacopo Staiano, Benjamin Piwowarski, Ewa Kijak, Vincent Claveau. "Choisir le bon co-équipier pour la génération coopérative de texte". Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles (TALN). 2022
- Antoine Chaffin, Julien Delaunay. "“Honey, Tell Me What’s Wrong”, Explicabilité Globale des Modèles de TAL par la Génération Coopérative. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN). 2023

Under review Finally, a work carried out during the thesis is still under review:

- Antoine Chaffin, Ewa Kijak, Vincent Claveau. "Distinctive Image Captioning: Leveraging Ground Truth Captions in CLIP Guided Reinforcement Learning". Under review. 2023

Show that ground truth captions, although not needed, can be very useful to train image captioning models using cross-modal retrieval model as reward in a reinforcement learning scheme. It is introduced in Chapter 10.

Big Science Alongside this work, I took part in the scientific consortium [Big Science](#) during my thesis which led to some publications. Although these will not be discussed in this manuscript, they are mentioned for completeness:

- Victor Sanh et al. "Multitask Prompt Tuning Enables Zero-Shot Task Generalization". Proceedings of the 2022 International Conference on Learning Representations (ICLR). 2022
- Big Science. "BLOOM: A 176B-Parameter Open-Access Multilingual Language Model". CoRR. 2022

PART I

GENERATING TRAINING DATA

TRANSFORMERS FOR NATURAL LANGUAGE GENERATION

Soon after the introduction of the Transformers [353] architecture, it became the standard for most machine learning tasks, including representation learning and natural language generation. Due to the increasing ubiquity of Transformers, we will focus on this architecture in this manuscript. In the next sections, we first introduce the Transformer architecture and its components, then we present its application to textual data and finally, we present how texts are generated using Transformers.

1.1 Transformers

1.1.1 Attention Layers

Transformers are built to process sequential data. It means that an input x is composed of a sequence of N tokens (x_1, x_2, \dots, x_N) . To process sequential data, RNN [288] and LSTM [142] process all sequence elements one after the other and keep a compressed representation of the sequence up to the considered token (a context vector). Because this representation has a fixed size, the more considered tokens the more compressed the information will be, reducing the performance of these models over long sequences. The attention mechanism [24] allows to dynamically compute the context vector c at timestep t through a weighted average of the representations $f(x_i)$ of other tokens in sequence:

$c_t = \sum_{i=1}^N \alpha_{ti} f(x_i)$. The weight of each token α_{ti} is computed using the softmax over the representations of each token processed by a feed-forward neural network trained to weight the alignment of the token i and t (compatibility function). Instead of blindly compressing the information at each step, the most important tokens at a given timestep are dynamically selected and their representations are fused to compute the representation of the current token. This allows to handle longer sequences and to retrieve information from earlier tokens that might have vanished in a progressive compression. The success of the Transformer architecture is attributed to this attention mechanism. The attention used in Transformers is called "scaled dot-product" attention because of the way the weights of the sum are computed. Attention layers compute three different matrices: Q (queries), K (keys) and V (values) using three learned projection matrices (feed-forward networks), W_Q , W_K and W_V respectively. After projecting the input sequence, the output of the attention layer is then computed by:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} = \mathbf{AV} \quad (1.1)$$

with d_k the projection dimension of Q and K . This represents, for a given token, the weighted average of values vectors of the sequence tokens, using the scaled dot product of its query vector with the key vectors of the sequence as weights. The layer computes, for a given token, which tokens in the sequence are the most compatible, and then merges the information from every token weighted by their compatibility. Figure 1.1 illustrates the attention mechanism for a text input, with the representation of a word in the next layer being computed by merging the representation of every word in the sentence weighted by their relationship with the given word (attention weights). The additional normalization term ($\sqrt{d_k}$) is applied to mitigate the vanishing gradient problem.

Unlike standard neural networks that have a fixed way to process different inputs (the weights of other tokens in the representation of a given token are learned once and for all), the attention layers allow to learn how to **dynamically** compute the weights of a given layer **depending on the input**. This can be seen as effectively instantiating one network adapted to the input among a whole range of other possibilities for processing the input sequence as well as possible. Note that, for a sequence of length N , attention layers require to compute N^2 attention weights (N attention weights per token), which induce a memory and computational complexity quadratic in the length of the input sequence. This is damaging when processing very long documents or using small atomic elements

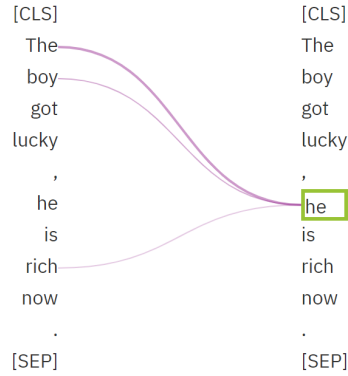


Figure 1.1 – Visualisation of the attention mechanism for the word “he” in one attention layer of a Transformer. The weights are computed using Equation 1.1, the higher the weight (darker color), the more the token influences the representation of the word in the next layer.

as tokens (such as characters or pixels). Sparse [177, 283], low-rank [66, 170, 366] and hybrid approximations [403, 60, 403] of the full quadratic attention have been proposed, but applying Transformers to very long sequences is still challenging [342].

Attention weights do not depend on the position, so attention layers are permutation invariant. Because there is also no recurrence or convolution, the model does not natively encode the notion of order of the input sequence, which can be damaging for example for text inputs where the order matters. Thus, positional encodings are added to the input embeddings to add information about the relative or absolute position of the token in the sequence. Several positional encodings can be used, whether learned or fixed [116]. In the original Transformer architecture, sinusoidal positional embeddings are used. Considering the original embedding of the token as a point in the embedding space, sinusoidal encoding creates a circular pattern around it, by moving the embedding to the same distance but with different directions. Tokens that are close in the input sequences receive similar perturbation while those that are far are pushed in a different direction.

To model even more relations in the input, and, by extension, create better representations, the attention mechanism is enhanced using **multi-head attention**. Multi-head attention is the process of running several attention layers in parallel. Each attention layer is composed of H heads and each head H^i will project the query, key and values matrices into different matrices at each layer using projection matrices \mathbf{W}_Q^i , \mathbf{W}_K^i and \mathbf{W}_V^i . Thus, Equation 1.1 actually computes attention weights over the input H times in parallel, significantly increasing the model capacity by creating multiple representation spaces. The

outputs of every head are then concatenated and projected by a last projection matrix W into the original input dimension:

$$\text{MultiHeadAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_H) \mathbf{W} \quad (1.2)$$

$$\text{where head}_i = \text{Attention}(\mathbf{Q} \mathbf{W}_Q^i, \mathbf{K} \mathbf{W}_K^i, \mathbf{V} \mathbf{W}_V^i) \quad (1.3)$$

This is helpful because a single head of attention only has a limited attention budget (because the softmax sums to 1), which limits the ability to attend to multiple positions without lowering the attention to some.

1.1.2 Encoder and Decoder

A Transformer is a composition of unitary elements, called Transformer *block*. Each of these blocks is composed of a multi-headed self-attention module followed by a feed-forward network. Self-attention refers to attention layers where queries, keys and values come from the same sequence, hence the sequence is attending to itself. This is in contrast with cross-attention layers that use keys and values from another sequence to compute the attention over the original sequence queries. While self-attention computes the compatibility of the tokens sequence with itself, cross-attention computes the compatibility with an external sequence and uses this external sequence to compute the representation of input tokens. These blocks are stacked on each other, projecting the input in an embedding space by processing the output of the previous block.

The original Transformer architecture is composed of two stacks of blocks: the encoder followed by the decoder. The encoder takes a sequence of input tokens and computes the sequence of vector embeddings representing the tokens. The decoder is then used to autoregressively generate an output sequence of tokens [384], that is, generating the output sequence one token at a time and conditioning the next one using previously generated ones. Cross-attention layers are added to the decoder to condition the generation process to the representation computed by the encoder. Indeed, Transformers have initially been introduced for Sequence-To-Sequence (Seq2Seq) tasks [333] where an input sequence is used to condition the desired target sequence. For example, for translation tasks, the generation of the sequence in the target language needs to be strongly conditioned on the sequence in the original language. An illustration of the encoder-decoder architecture is given in Figure 1.2. Also, layer normalizations [22] and residual connections [134] are

applied around attention layers and feed-forward networks to avoid vanishing gradient.

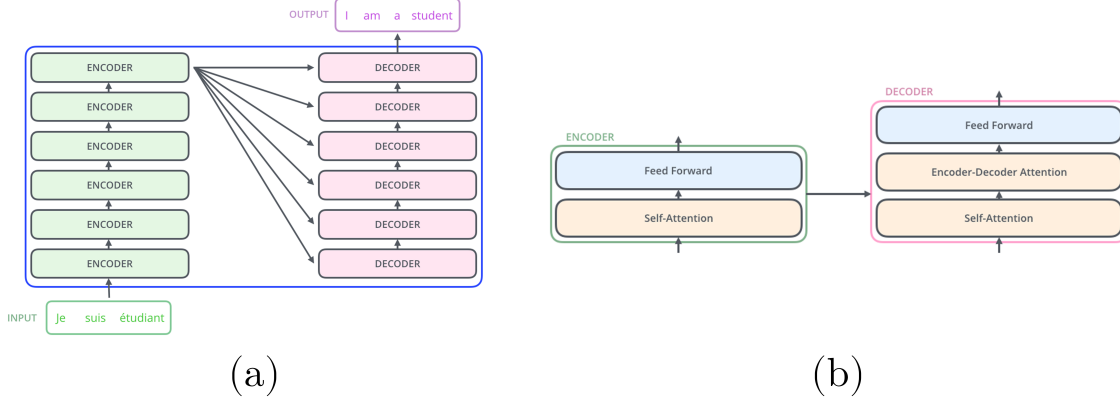


Figure 1.2 – Illustration of the original Transformer encoder-decoder architecture used for a translation task (a) and the different components of the blocks composing the two stacks (b). Illustration taken from the [Illustrated Transformer](#).

However, self-attention layers prevent the learning of the auto-regressive objective. Indeed, since in self-attention layers every token can attend to every token in the sequence, a token could "see" the token to be produced, leaking the target information. Hence, to learn to generate tokens auto-regressively, a causal mask needs to be applied to make the self-attention unidirectional, preventing the decoder from attending to future tokens when trying to predict them. The causal mask is applied by setting the result of the product of the query and the keys of the next tokens to $-\infty$, effectively zeroing the contribution of tokens that appear later in the sequence through the softmax operation (i.e., $\alpha_{ti} = 0, \forall i > t$). In practice, an upper-triangular matrix M with dimension $\mathbb{R}^{n \times n}$ is multiplied element-wise to the result of \mathbf{QK}^T to define masked self-attention:

$$\text{MaskedAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{d_k}} \circ \mathbf{M}\right)\mathbf{V} \quad (1.4)$$

with \circ denoting the Hadamard product.

Although the original Transformer is composed of an encoder followed by a decoder (encoder-decoder), it is possible to use a single stack alone (decoder-only or encoder-only). The only difference in this setup is that an encoder leverages bidirectional self-attention, whereas a decoder mask restricts the attention unidirectionally. An illustration of these different attention schemes is given in Figure 1.3.

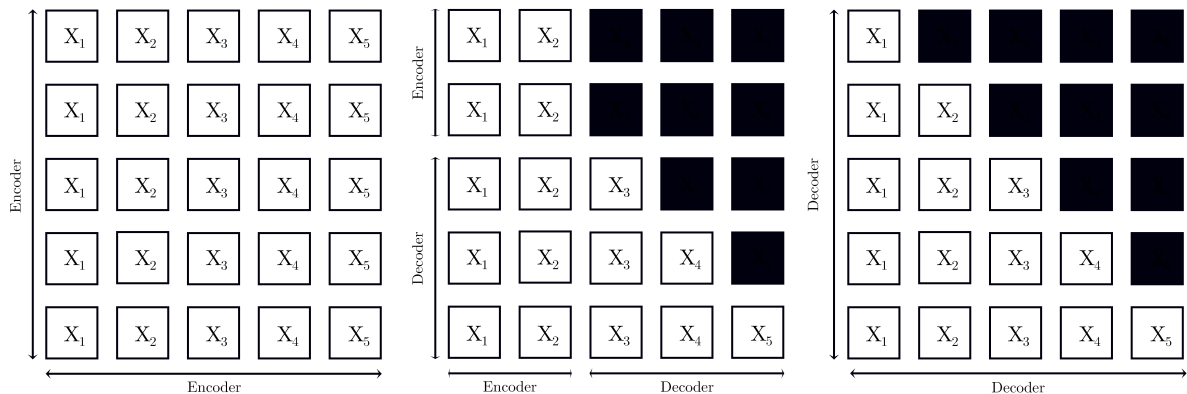


Figure 1.3 – Representation of the different attention masks used in the different architecture of Transformers, from left-to-right: encoder-only, encoder-decoder and decoder only. Black cells represented masked tokens, that is, tokens not attended to compute the representation of the token.

1.1.3 Pre-training

The Transformer architecture has become the standard in deep networks because of its outstanding results. These results have been achieved thanks to the very large capacity offered by the attention layers but also to their nice scalability leveraged by gigantic pre-training on very large datasets. **Pre-trained models** refers to models that have gone through extensive and expensive training and are then used as off-the-shelf features extractors or as a starting point for a subsequent training, referred as fine-tuning. This paradigm particularly fits Transformers, because their strong capacity to model complex and arbitrary relations within the input sequence coupled to their very large parameters count make them prone to overfitting on modestly-sized datasets [268, 353, 126]. Even though dropout [326] is applied to every sub-layers (feed-forward network, skip connection and attention layers) and at the input and output of a stack to prevent overfitting, Transformers require very large training datasets. Building large-scale labeled dataset is challenging and very costly, while constructing large-scale unlabeled corpora is more straight-forward. Hence, the most heavy pre-training phase is done on unlabeled data using diverse **self-supervised training** objectives to capture knowledge and create representations from very large corpora. Self-supervised learning is used to automatically extract pseudo-labels used for a pretext task that makes the model learn a useful representation for the downstream tasks. It allows to learn from the vast amount of unlabeled data available such as [Common Crawl](#). In addition to learning universal representations and providing a

good initialization, this heavy unsupervised training effectively acts as a regularizer when later fine-tuning the model on small datasets [96]. Although leveraging the abilities of strong pre-trained models predates Transformers [229, 139, 74, 228, 258, 261], their scalability really pushed this idea at its peak, now resulting with billion-parameters models trained on internet-scale datasets and directly used for a large set of tasks in a zero-shot fashion (i.e., without any task-specific fine-tuning).

1.2 Textual Transformers

Transformers are adapted to textual data because of their sequential nature: a sentence corresponds to a sequence composed of its (sub-)words. The text is first tokenized, for example using Byte-Pair Encoding (BPE) [112] or SentencePiece [184], and the resulting sequence of tokens are then fed to the Transformer as a sequence of one-hot vectors corresponding to the tokens ids. These ids are then projected into an initial static embedding layer, that is a lookup table similar to Word2Vec embeddings, to obtain a continuous representation before going into Transformer layers.

1.2.1 Pre-training Objectives

As previously introduced, a lot of expensive labeled data [167] are required to successfully train deep networks using the supervised cross-entropy and it results in models that struggles to generalize beyond the training data [341, 275]. Unsupervised training frees from the need for labelling data, removing the limiting factor for scaling deep networks and limiting annotator biases. Thus, textual Transformers are first pre-trained using self-supervised objectives on very large text corpora without any human annotation before being tuned on task-specific data. During this pre-training, the model learns a general representation space of texts based on the distribution of the data alone.

Different objectives can be used, depending on the downstream task and model architecture (encoder-decoder, encoder only or decoder only). The most straight-forward (and original objective) is language modeling. Language modeling estimates the probability distribution of sequences of symbols x_1, x_2, \dots, x_T (most often tokens) taken from a vocabulary \mathcal{V} , with variable lengths T . The model is trained to estimate the probabilistic distribution of the data by maximizing the probability of training sequences. To do so, the probability of one target sample x (also called *likelihood*) is defined as the

joint probability over each of its tokens, which can be factorized using the chain rule: $p(x_{1:T}) = \prod_{t=1}^T p(x_t | x_{1:t-1})$. The weights θ of a Transformer are thus optimized using a language modeling head, a layer that outputs a probability distribution over the vocabulary for the next token given the input ones, i.e. $p_\theta(x_t | x_{1:t-1})$ at a given time step t [34]. As only the exact Ground Truth (GT) sequence is guaranteed to be right and the correctness of even very small variations of it is unknown, the model is trained to predict the next token from the GT x^{gt} given previous GT tokens, by optimizing its probability through a cross-entropy loss between the one-hot representation of the target token and the output probability distribution. This defines the **Teacher Forcing** (TF) loss [381]: $L_\theta(x^{gt}) = -\sum_{t=1}^T \log p_\theta(x_t^{gt} | x_{1:t-1}^{gt})$. This results in a Maximum Likelihood Estimation (MLE) training that maximizes the log-likelihood of the token sequences included in training samples under the model. The model can then generate text by iteratively completing an initial sequence, called the *prompt* (see Section 1.3). This objective is used to train open-ended generation models, but also dialog, summarization, question answering and machine translation, using adapted prompts.

Another training objective is **Masked Language Modeling** (MLM), first introduced as the Cloze task [344] and adapted to a pre-training task by [88]. Rather than predicting the next token given previous ones, the model is tasked to predict tokens that have been masked in the input sequence from the rest of the tokens. The loss becomes, for a set of masked tokens \tilde{X} , $L_\theta(x) = -\sum_{\tilde{x} \in \tilde{X}} \log p_\theta(\tilde{x} | x_{\setminus \tilde{X}})$. Note that, contrary to standard language modeling, the model can use information from tokens to the right of the token to be predicted, creating a bidirectional representation. However, this objective introduces a mismatch between the pre-training and fine-tuning tasks, because the model is not exposed to the [MASK] token during fine-tuning or inference. To solve this issue, authors originally proposed to use the [MASK] token 80% of the time and to either replace the token by a random one or keep it unchanged the rest of the time when masking. This objective is an instance of denoising autoencoders, because the model is tasked to reproduce the input after projecting it in a low dimensional space, as in autoencoders, but with an additionally corrupted input. MLM can be seen as a classification problem, with the label being masked token id, and so can be solved by an encoder by feeding its embedding to a classification head [88]. This can also be seen as a Seq2Seq problem, where the decoder generates the masked tokens [271]. Seq2Seq MLM enables other ways to corrupt the text for pre-training [194] such as token deletion (the model is tasked to predict positions of missing tokens), text infilling (whole spans are masked rather than single tokens), sentence

permutation (model needs to put the sentences back in order) or document rotation (finding the start of a rotated document). Note that attention masks can be used to train a unique model for different type of MLM (unidirectional, bidirectional and sequence to sequence) [90, 29].

1.2.2 Architectures

When it comes to discrimination, models based on bidirectional attention are commonly used since "intuitively, it is reasonable to believe that a deep bidirectional model is strictly more powerful than [...] a left-to-right model" [88]. Encoder-only architectures such as BERT [88] and RoBERTa [209] have thus been prevalent in natural language understanding tasks such as classification. For sequence-level discriminative tasks, the common practice introduced by BERT is to prepend a special [CLS] token to the input sequence and to use the embedding of this token to perform the task, for example by feeding it to a Multilayer Perceptron (MLP) to do classification. The idea is that attention layers enable the information to flow between all the tokens towards this token that serves as a catalyst and contains the information relevant for the task. However, while the bidirectionality brings some capacity to the model, it also makes it non-auto-regressive: when a token is added at the end of a sequence, it changes the representation of previous tokens and every hidden state needs to be re-computed. This also prevents to train the model using next token prediction (teacher forcing) because the information of the token to predict would leak. Thus, decoder-only, such as GPT models [268, 269, 46], have been used for language modeling tasks.

While this distinction between discriminative and generative tasks is intuitive, most text processing tasks can be cast into a sequence-to-sequence problem [271]. For example, the model can be trained to generate the target label in classification tasks. The probability associated to labels tokens is then used as the classification probability. For named entity recognition, rather than classifying each token to entity or non entity, the model can be trained to produce span containing the entities. This allows to unify the different Natural Language Processing (NLP) tasks to a same objective and so to train an unique model to a large variety of tasks that will leverage the knowledge from one task and transfer it to another one. Encoder-decoder such as T5 [271] and BART [194] leverages the encoder to strongly constrain the downstream decoder using a task-specific prefix and are able to perform both natural language understanding and generation tasks. Note that, although an encoder-only cannot be used in this setting (because it can not be trained for generative

tasks), a decoder-only can also be used by feeding the prefix as the prompt of the decoder. The model thus learns to generate the answer while being conditioned on the prompt, but without the benefit of bidirectional attention within the prompt. A variant of the causal mask can be used to enable bidirectional attention on the prefix in a decoder, resulting in a non-causal decoder-only architecture [208, 271] that offers a good trade-off [367] over using an external encoder by sharing weights while allowing strong modeling on the prefix. This is especially useful for In-Context Learning (ICL), where the description and demonstrations of an unseen task is fed into the prompt to make the model "learn" the task without gradient descent, during the inference. The unification of NLP tasks under a same definition is very useful for multitask training of a model [269], which, in addition to use an unique model to solve multiple tasks, gives language models the ability to generalize to unseen datasets and tasks [294, 376].

1.2.3 Transfer Learning

Besides ICL and zero-shot applications of the pre-trained model, transferring the knowledge learned during pre-training to the target task – **transfer learning** [251] – is an open question. The pre-training tasks and datasets as well as the architecture of the model obviously impact the performance depending on the downstream task. Ideally, the model should have a viable architecture for the task (e.g., (encoder-)decoder for generative tasks), being pre-trained on data whose domain and distribution are close to the ones of the target data, and a pre-training objective that helps to build useful representations for the task. Then, the knowledge of the model can be transferred either by using it as a frozen feature extractor used as input to train an external model or fine-tuning it on the target task.

When using the model as an off-the-shelf feature extractor, it is important to choose the right layer, as the latest are closer to the pre-trained tasks, while the earliest are close to the static embedding layer [260, 261, 345]. Being too close to a given task might be damaging if useful information for the downstream task are not used to solve the pre-training task, while static embeddings might fail to capture higher-level of information.

While extracting features from a pre-trained model can be useful for tasks that require a specific architecture, the versatility of Transformers allows to fine-tune them for most of the NLP tasks. Simply fine-tuning the model on the target task is a convenient and general way to transfer the knowledge acquired during pre-training. If the original pre-training domain is too far from the target one, an additional pre-training can be done before the final tuning.

This two-stage fine-tuning allows a smoother adaptation and increases the performance on the downstream task [331, 263, 113, 202]. To perform this fine-tuning step, adapted data is required. This data should be as representative of the downstream task as possible, so the model learns to correctly solve it. As detailed in the introduction, in the context of multimodal misinformation detection, creating a representative dataset is challenging. Thus, in this thesis, we explore the use of generated texts to train discriminative models when collecting organic data is not possible or difficult.

1.3 Decoding Methods for Natural Language Generation

As introduced in the previous section, the teacher forcing loss is used to train a neural network to output a probability distribution over the vocabulary for the next token given the input ones. This allows to create an auto-regressive language model (LM) that can generate sequences by iteratively using those distributions to emit a token x_t , and append it to the context $x_{1:t-1}$ for the next iteration. The generation process –or *decoding*– is started using a small initial sequence: the prompt.

However, selecting a token x_t so that the final sequence has the highest possible likelihood is still an open problem. This is mainly due to the size of the search space, since for a sequence of length T and a vocabulary of size $|\mathcal{V}|$ (often being tens of thousand), the space has a size of $|\mathcal{V}|^T$, making an exact search intractable. Several search methods have been proposed to alleviate this issue; a naive one is the **greedy search** where the most probable token is picked at each step. In order not to miss a more probable sequence which starts with a token having a lower probability, **beam search** [85] explores the k (number of beams) most probable sequences at each generation step.

It should be noted that these search methods are highly biased towards samples that have a high likelihood in the beginning, which does not guarantee to find the sequence with the highest possible likelihood. Moreover, because only the most probable words are chosen, they tend to get stuck in repetition loops and generate redundant texts [389]. This phenomenon can be mitigated by randomly sampling from the distribution. However, sampling in the full distribution results in an increasingly high probability of picking a very unlikely token in the very long tail of the distribution that will break the generation process. Thus, **top-k sampling** [100] samples the next token from the k most probable next tokens to introduce variance while removing the long tail. However, k is fixed and the k^{th} token may have a very low probability (and therefore be off-topic). To improve

upon this method, **nucleus sampling** [144] (or **top-p sampling**) is used to sample from the smallest set of tokens that have a cumulative probability higher than p . Finally, **beam sampling** [49] consists in a mix of beam search and sampling. k tokens are first sampled using the LM distribution and, for each one, k more tokens are then sampled, leading to k^2 beams in which only the k most probable are kept. This process is repeated until each beam is finished.

Note that a neural LM actually predicts *logits* $z_{1:|\mathcal{V}|}$ over the vocabulary \mathcal{V} . The probability of sampling the i^{th} token at time step t is computed by applying softmax, usually with temperature τ :

$$p_{\theta}(x_i \mid x_{1:t-1}) = \frac{\exp(z_i/\tau)}{\sum_v \exp(z_v/\tau)} \quad (1.5)$$

The temperature parameter influences the entropy of the distribution; a high temperature flattens the distribution, leading to a uniform distribution, and a low temperature creates a high peak on the most probable token. This defines a trade-off between generating diverse and coherent samples.

GENERATING ARTIFICIAL TEXTS AS SUBSTITUTION OR COMPLEMENT OF TRAINING DATA

Contents

2.1	Data Augmentation for Natural Language Processing	33
2.2	Generating Artificial Data	35
2.2.1	Fine-Tuning the Language Model	36
2.2.2	Text Generation	36
2.3	Experiments	37
2.3.1	Experimental Setting	37
2.3.2	Neural Classification	40
2.3.3	Bag-Of-Words Classification	42
2.4	Conclusion	44

In this first chapter, we will start by exploring the use of artificially generated texts for supervised machine learning tasks within two different scenarios: using the artificial data as a complement of the original training dataset (for instance, to yield better performance) or using it as a substitute of the original data (for instance, when the original data cannot

be shared because they contain confidential information [10] or because of intellectual property in the case of news articles). Although this setup is different from our target one, as discussed at the end of the chapter, it is a preliminary experiment that checks if generated samples are helpful for training. At the time of the experiments, it was not as obvious as it might be today. Besides, it is also a good starting point to understand the potential of generated texts and their issues.

The generation of these artificial texts is performed with a neural language model trained on the original training texts. We show the interest of these scenarios with several text classification tasks, handling well-written or noisy language: fake news detection, opinion mining and news categorization. We further explore the interest of these scenarios with different classifiers, including simple and explainable classifiers based on bag-of-words representation. This explainable property, as discussed later, is particularly useful for misinformation detection models, to detect biases and prevent harmful processing, as well as help human fact-checkers.

Precisely, the main research questions studied are the following ones:

1. what is the interest of text generation to improve text classification (complement);
2. what is the interest of text generation to replace the original training data (substitution);
3. what is the interest of text generation for explainable classifiers, based on bag-of-words representation.

After a presentation of usual data augmentation techniques for natural language processing in Section 2.1, we detail our data augmentation scenarios based on artificial text generation (Section 2.2). The tasks and experimental data are described in Section 2.3.1. The experiments and their results for each of our research questions are reported in Section 2.3.2 for the neural classifiers and Section 2.3.3 for bag-of-words based classifiers.

Personal contribution This work arises from the ideas and code of my internship preceding my thesis. I also took part in the experiments and the redaction once my thesis started.

2.1 Data Augmentation for Natural Language Processing

Data augmentation refers to methods that generate synthetic data from existing data to augment the training dataset. This is helpful for different reasons, such as alleviating

data scarcity and limiting overfitting. Note that the new data should be as close as possible to the original data with respect to the task (e.g., same class). Although less extensive than for computer vision tasks [312], there are several data augmentation techniques for tasks of NLP.

Some approaches add noise to the original examples to make the model more robust. The noise comes from more or less complex automatic perturbations of the original examples, similarly to traditional computer vision augmentations. For example, swapping the order of two words [375] or longer spans such as sentences [391]. The addition or deletion of words [375] or sentences [391] is another source of noise. Then, some words can be replaced, either by random ones [371] or by synonyms [178, 375, 236, 166] to keep the original semantics as much as possible. The synonyms are found in external resources (such as WordNet [230]) [410, 375], or computed from static word embeddings [370] (such as Glove [258] or word2vec [228]). Some studies leverage masked language models (such as BERT [88]) to perform the local modification [386, 162]. These approaches work by masking a word in the original example and using the model to replace it based on the context. This resolves the possible ambiguity that might face static embeddings. It is worth noting that, contrary to our proposed approach, when replacing a single word, the new example is not totally different (the syntactic structure of the new example is for instance identical to the original one).

Auto-regressive language models can also be used for data augmentation. For example, through back-translation [387, 395, 99], that is, translating the original text to another language and then translating it back to the original language. This results in a modified version of the input sequence without replacing individual words. Even closer to our study, some works use auto-regressive language models to produce a large quantity of data that are similar to the original data distribution for relation extraction [252], sentiment analysis of critics and questions [185] or for the prediction of hospital readmission and phenotype classification [10]. These studies, contemporary to this one, suggest that neurally generated samples can be used as data augmentation. In addition to adding more explorations on this subject, our interest here is also to examine the gains and losses of our different scenarios of using artificial data, their preparation, and to examine their effects on different families of classifiers.

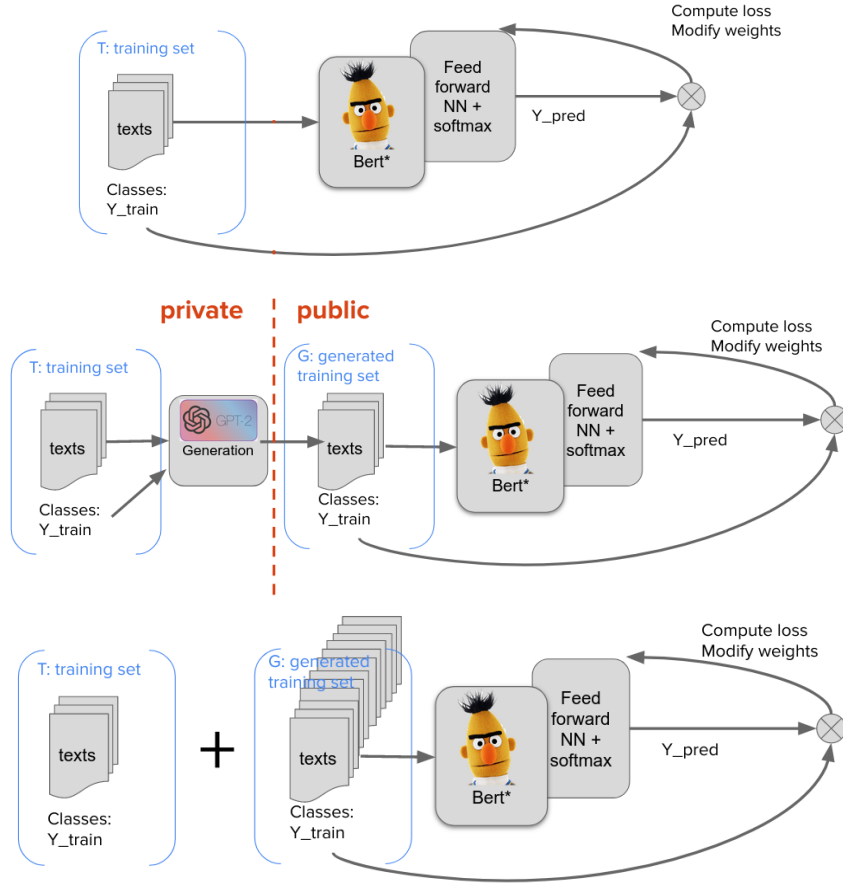


Figure 2.1 – Different training scenarios: usual text classification framework (here with a standard BERT classifier), generated data used as substitution (especially useful when the original data cannot be shared), generated data as complement.

2.2 Generating Artificial Data

Let us assume to have a set of original texts \mathcal{T} divided into n classes c_i , from which we wish to generate artificial texts \mathcal{G}_{c_i} for each class c_i . As explained in the introduction, we want to examine different scenarios of usage of these generated data: complement or substitution. The scenarios, as well as the usual text classification framework, are exemplified in Figure 2.1.

We use GPT-2 [269] because the experiments reported below need fine-tuning and, at the time of the experiments, it was one of the most performant generative language models whose weights were available. Better models released since should yield better results, but the experiments reported below are still relevant to study the interest of the proposed

scenarios. GPT-2 is trained on more than 8 million documents from Reddit (i.e., general domain language such as discussions on news articles, mostly in English).

2.2.1 Fine-Tuning the Language Model

The first step is to fine-tune a language model on the original texts \mathcal{T} , so it generates artificial texts that have the features of a target class c_i . For this fine-tuning step, we start from the large model (774M parameters) pre-trained for English and made available by [OpenAI](#). We fine-tune one language model per class with the original training data from this class \mathcal{T}_{c_i} . Another possible training procedure is to adapt a single model but to condition it with a special token indicating the expected class at the beginning of the text sequence (i.e., at the beginning of each original example), as done by CTRL [172]. Due to the limited amount of data available per class (compared to the number of parameters of the model), we limit the number of steps to 2000 to avoid overfitting. The other fine-tuning parameters are the default ones of the OpenAI GPT2 code that is used in our experiments. On a Tesla V100 GPU card, this fine-tuning step lasts about 1 hour for each dataset (see below).

2.2.2 Text Generation

For each class c_i of the dataset \mathcal{T} , we use the corresponding model to generate artificial texts \mathcal{G}_{c_i} which hopefully will fall into the desired class. We provide prompts for these texts in the form of a start-of-text token followed by a word randomly drawn from the set of original texts. For the generation, we used the default values that we give here for reproducibility purposes, without detailing them (see the GPT-2 documentation): temperature = 0.7, top_p = 0.9, top_k = 40. The generation of examples relies on <https://github.com/minimaxir/gpt-2-simple>.

The texts generated for the class c_i containing a sequence of 5 consecutive words appearing identically in a text of \mathcal{T}_{c_i} are removed. This serves two purposes : on the one hand, it limits the risk of revealing an original document in the case where the \mathcal{T}_{c_i} data are protected, and on the other hand, it limits the duplicates which are harmful to the training of a classifier in the case where the \mathcal{G}_{c_i} data are used in addition to \mathcal{T}_{c_i} . In practice, it concerns about 10% of the generated texts in our experiments. In the experiments reported below, 16,000 texts are generated for each c_i class (this number of texts has been fixed arbitrarily).

In the scenario where the data cannot be distributed, it is appropriate to ask whether sensitive information can be recovered with the proposed approach. If the whole generative model is made available, this risk has been studied [53], and exists, at least from a theoretical point of view under particular conditions¹. Providing the generator itself is thus not possible. When only the generated data are made available, there is also a risk of finding protected information in them. Without other safeguards, it is indeed possible that among the generated texts, some are paraphrases of sentences of the training corpus. However, in practice, the risk is very limited:

- first of all, because there is no way for the user to distinguish these paraphrases among all the generated sentences;
- secondly, because additional measures can be taken upstream (for example, de-identification of the training corpus) and downstream (deletion of generated sentences containing specific or nominative information...);
- Finally, more complex systems to remove paraphrases, such as those developed for the Semantic Textual Similarity tasks [161, *inter alia*], can even be considered.

These measures make it highly unlikely that any truly usable information can be extracted from the generated data.

2.3 Experiments

2.3.1 Experimental Setting

The experiments detailed in the next section are diverse classification tasks: fake news detection in tweets, sentiment analysis in reviews and news article categorization. To study if the proposed approach works well across different language, the fake news and news categorization datasets consists of tweets/texts in English while the sentiment analysis dataset is in French.

FakeNews MediaEval 2020: English dataset of tweets This dataset was developed for the detection of fake news within social networks as part of the MediaEval 2020 FakeNews challenge [264]. Tweets about 5G or coronavirus were manually annotated according to three classes $c_i, i \in \{'5G', 'other', 'non'\}$ [298]. '*5G*' contains tweets propagating conspiracy theories associating 5G and coronavirus, '*other*' are for tweets propagating other conspiracy

1. See also the discussion on the Google AI blog: <https://ai.googleblog.com/2020/12/privacy-considerations-in-large.html>.

- If the FBI ever has evidence that a virus or some other problem caused or contributed to the unprecedented 5G roll out in major metro areas, they need to release it to the public so we can see how much of a charade it is when you try to downplay the link.
 - So let's think about this from the Start. Is it really true that 5G has been activated in Wuhan during Ramadan? Is this a cover up for the fact that this is the actual trigger for the coronavirus virus? Was there a link between 5G and the coronavirus in the first place? Hard to say.
 - We don't know if it's the 5G or the O2 masks that are killing people. It's the COVID19 5G towers that are killing people. And it's the Chinese people that are being controlled by the NWO

Figure 2.2 – Examples of tweets artificially generated with a GPT-2 model trained on the MediaEval examples with class \mathcal{T}_{5G} .

theories (which may be about 5G or covid but not associated), and '*non*' tweets not propagating any conspiracy theories.

It is worth noting that the classes are imbalanced; indeed, in the training dataset \mathcal{T} : $|\mathcal{T}_{5G}| = 1,076$, $|\mathcal{T}_{\text{other}}| = 620$, $|\mathcal{T}_{\text{non}}| = 4,173$.

The data augmentation (i.e., text generation) is performed as explained in the previous section. Figure 2.2 presents three examples of generated texts from the MediaEval 2020 '5G' class.

AG_news: news classification [AG_news](#) is a large collection of news articles in English, used for different NLP tasks. Hereafter, we use it as a classification dataset, as proposed by [410]. In this setting, the task to perform is a thematic classification of article title and associated snippets dataset into 4 categories: (**world**, **sport**, **business** and **sci/tech**). We use data and the train/test split as provided by [HuggingFace datasets](#) in which the classes are balanced. The generation is done similarly to the previous dataset, by fine-tuning an English GPT-2 large model on the training examples.

FLUE CLS-FR: French dataset for sentiment analysis The third dataset is taken from the FLUE evaluation suite for French [189]. It is the French part of the Cross Lingual Sentiment (CLS-FR) dataset [266], which consists of product reviews (book, DVD, music) from Amazon. The task is to predict whether the review is positive (rated more than 3 stars on the merchant site) or negative (less than 3 stars). The dataset is divided into balanced training and test sets. In our experiments, we do not distinguish between products : we have only two classes (positive and negative) with reviews of books, DVDs or musics.

- Déçue... J'ai eu je l'avoue du mal à lire ce livre arrivé au milieu de celui-ci. L'histoire ne paraît pas vraiment très réaliste. Le policier est plus guidé par de la chance que par son instinct. Que se serait-il passé s'il n'avait pas rencontré cette dame insolite ? Non ! Je ne crois pas que je lirais d'autres livres de fred vargas... Dommage je n'ai pas encore trouvé une source infaillible de bons polars.
translation: Disappointed... I have to admit that I had a hard time reading this book until the middle of it. The story doesn't really seem very realistic. The policeman is guided more by luck than by his instinct. What would have happened if he had not met this unusual lady? No! I don't think I would read any more books by fred vargas... Too bad I haven't found an infallible source of good thrillers yet.
- De la daube. Cet homme ferait mieux de mettre son piano à la benne. Il n'y a pas de musicalité, ce disque irrite et agresse, ou au mieux il agresse et abuse son timbre et pénible accent amoureux. Musicalement, c'est de la musique de... chandler, on se dit... "c'mere irons up". Une chose est sûrement restée disponible sur cet album, mais attention aux maisons de disque !
translation: Rubbish. This man would do better to put his piano in the garbage. There is no musicality, this record irritates and assaults, or at best he assaults and abuses his timbre and painful love accent. Musically, it is music of... chandler, we say to ourselves... "c'mere irons up". A thing surely remained available on this album, but attention to the record companies!
- Gros navet. Décor atrocement kitsch, couleurs d'un mauvais goût abominable qui rendrait effleuré un ami en le dire... ça marche. Aucun suspense, tout est répétitif, les personnages sont inconséquents, ennuyeux. A éviter absolument.
translation: Such a turkey. Atrociously kitsch decor, colors of an abominable bad taste that would make a friend shudder to say it. No suspense, everything is repetitive, the characters are inconsistent, boring. To avoid at all costs.

Figure 2.3 – Examples of artificial reviews generated with a GPT-2 model trained on the CLS-FR examples with the class $\mathcal{T}_{\text{negatif}}$.

As with the MediaEval data, a language model is tuned for each class using the training data. Generation is then done as described in the previous section. Examples of generated negative reviews are given in Figure 2.3.

As can be seen from these examples (including the MediaEval examples in Figure 2.2), the generated texts seem to belong to the expected class and exhibit distinctive features (see Section 2.3.2 for a discussion of this point). However, they often have flaws that make the fact that they were generated detectable. This is particularly the case for French texts, which can be explained by the fact that we did not have, at the time of the experiments, a pre-trained model for French; the model, as well as the tokenizer, are therefore based on the English GPT model. Using natively pre-trained models for French could improve this aspect.

2.3.2 Neural Classification

In the experiments reported below, the performance is measured in terms of micro-F1 (equivalent to accuracy), and, to take into account the imbalance of the classes (in the MediaEval dataset), in terms of macro-F1 and MCC (Matthews Correlation Coefficient), as implemented in the library [scikit-learn](#) [256]. The performance is measured on the respective official test sets of the MediaEval [264] and CLS-FR [189] tasks, of course disjoint from the training sets \mathcal{T} .

First results For our first experiments, we use state-of-the-art neural classification models based on Transformers. For the MediaEval data, in English, we opt for a RoBERTa [209] pre-trained model for English (large model with a classification layer). It is this type of Transformer-based model that obtained the best results on these data during the MediaEval 2020 challenge [59, 73]. Among the variants of BERT [88], RoBERTa was preferred here for its tokenizer that is more adapted to the specifics of the very free form of writing found in tweets (mix of upper and lower case, absence or multiplication of punctuation, abbreviations...). For the CLS-FR data of FLUE, we use the large-cased FlauBERT model [189]. This allows us to compare with the results originally published on these data. For both models, we use the implementation of the [HuggingFace’s Transformer library](#) [382]. The batch size is set to 16 and the number of epochs set to 3 in all scenarios (optimal number of epochs for the baseline), except for the last one (3 on \mathcal{G} followed by 1 on \mathcal{T}).

We evaluate the performance according to our different training scenarios: on the original data \mathcal{T} (which serves as a baseline), on the artificial data \mathcal{G} , and finally on both the artificial and original data. In this last case, we test two training strategies :

- the first, $\mathcal{T} + \mathcal{G}$, mixes the original and artificial examples,
- the second, \mathcal{G} then \mathcal{T} , trains on the artificial data on the first epochs, then on the original data for the last epoch. This results in a kind of fine-tuning on the original data after a first training on the artificial data.

The results for the MediaEval and CLS-FR datasets are reported in Table 2.1. On the CLS-FR data, we observe very few differences between the different scenarios and the baseline (note that our baseline is similar to the published state-of-the-art results). This classification task is relatively simple because the two classes (positive and negative) have very distinctive and identifiable lexical fields. This makes it possible to generate artificial data of as good quality as the original data, leading to comparable results, even without

model	MediaEval			CLS-FR			AG_news		
	micro-F1	macro-F1	MCC	micro-F1	macro-F1	MCC	micro-F1	macro-F1	MCC
BERT* / \mathcal{T}	79.57	62.66	55.71	95.44	95.42	90.86	94.35	94.35	92.47
BERT* / \mathcal{G}	62.68	54.03	39.27	95.13	95.12	90.25	90.25	90.25	87.12
BERT* / $\mathcal{T} + \mathcal{G}$	75.01	58.81	46.37	95.43	95.42	90.89	92.44	92.44	88.51
BERT* / \mathcal{G} then \mathcal{T}	79.89	60.64	52.02	95.76	95.75	91.51	94.88	94.88	92.57

Table 2.1 – Performance (%) of neural classification approaches on the MediaEval, CLS-FR, and AG_news tasks according to the scenarios of usage of the artificially generated texts without filtering. The BERT* model are respectively RoBERTa, FlauBERT and RoBERTa.

training on the original texts. On this type of task, artificially generated data can therefore be used without loss of performance.

The MediaEval task is more difficult as can be seen with the results of the baseline (RoBERTa / \mathcal{T}). On these data, in a substitution scenario (i.e., when the generated data are used alone as training data), the results are strongly degraded compared to a system trained on the original data. This is of course because the data generated by each of the language models may not belong to the expected class, as the models do not fully capture the more fine-grained specificity of the fine-tuning data. In a scenario where we aim at augmenting the training data, the impact is less significant, especially if the artificial data is used only in the first few epochs.

Results with automatic filtering Indeed, the examples generated by our trained GPT-2 models \mathcal{G} may contain texts that do not belong to the expected classes. Manually filtering or annotating these texts is of course possible but remains a costly task. To reduce the effect of these texts on the classification at a low cost, we propose to exclude them using a classifier learned on the original data \mathcal{T} . Its goal is to filter the generated examples: any text of \mathcal{G}_{c_i} which is not classified c_i by the classifier is removed. In this way, we hope to eliminate, automatically, the most obvious cases of problematic artificial texts. In the following experiments, we use the RoBERTa classifier trained on \mathcal{T} (evaluated in the first row of Table 2.1). In this way, 40% of the examples are removed. The resulting filtered artificial dataset is noted \mathcal{G}^f .

The results with these new filtered sets of artificial examples in the same training scenarios are presented in Table 2.2 for the MediaEval and CLS-FR data. It can be seen that this filtering strategy pays off, with improved performance on all metrics compared to no filtering. In the substitution scenario, the performance is now close to the baseline, and is even better on the macro-F1; this is explained by the fact that the artificial set \mathcal{G} is

model	MediaEval			CLS-FR			AG_news		
	micro-F1	macro-F1	MCC	micro-F1	macro-F1	MCC	micro-F1	macro-F1	MCC
BERT* / \mathcal{T}	79.57	62.66	55.71	95.44	95.42	90.86	94.35	94.35	92.47
BERT* / \mathcal{G}^f	76.22	64.18	52.75	95.76	95.75	91.51	93.49	93.49	91.35
BERT* / $\mathcal{T} + \mathcal{G}^f$	80.12	66.08	57.44	95.99	95.98	91.97	93.47	93.47	91.34
BERT* / \mathcal{G}^f then \mathcal{T}	83.55	67.90	60.05	95.96	95.95	91.96	95.10	95.10	92.89

Table 2.2 – Performance (%) of neural classification approaches on the MediaEval, CLS-FR, and AG_news tasks according to the scenarios of usage of the artificially generated texts after filtering. The BERT* model are respectively RoBERTa, FlauBERT and RoBERTa.

much more balanced than \mathcal{T} and thus performs better on the minority classes of the test set. In the complement scenario, we observe a significant improvement over the baseline, especially with the sequential strategy \mathcal{G}^f then \mathcal{T} .

Differences between classifiers Beyond the global performance measures, it is interesting to check if the classifier trained on the artificial data allows to make the same decisions as a classifier trained on \mathcal{T} . To do so, we can look at the proportion of examples (from the test set) for which the decision of BERT* / \mathcal{T} and BERT* / \mathcal{G}^f differs. For the CLS-FR data, the classifiers agree on a large majority of examples. Figure 2.4 shows the confusion matrix of FlauBERT / \mathcal{T} and FlauBERT / \mathcal{G}^f on the CLS-FR data.

From this confusion matrix, we can see that the classifiers do agree on the majority of examples. The cases of disagreement are proportionally more important on the false positives and false negatives, but even for these categories, we still find a lot of common errors (42 and 77 examples respectively for the false positives and false negatives). The classifiers have therefore not only comparable performance but also very similar behaviors in detail since they give the same class on most examples.

2.3.3 Bag-Of-Words Classification

We also tested classifiers based on bag-of-words representations; we present only the results of the logistic regression (LR) which gave the best results. In general, these classifiers perform worse than the Transformer-based approaches, but they allow for better explainability [231, 54], for example by examining the regression weights associated with words. They are also way less expensive to train.

First results The texts are vectorized with tf-idf weighting and L2-normalized using scikit-learn. The parameters of the logistic regression are the default ones except for the following:

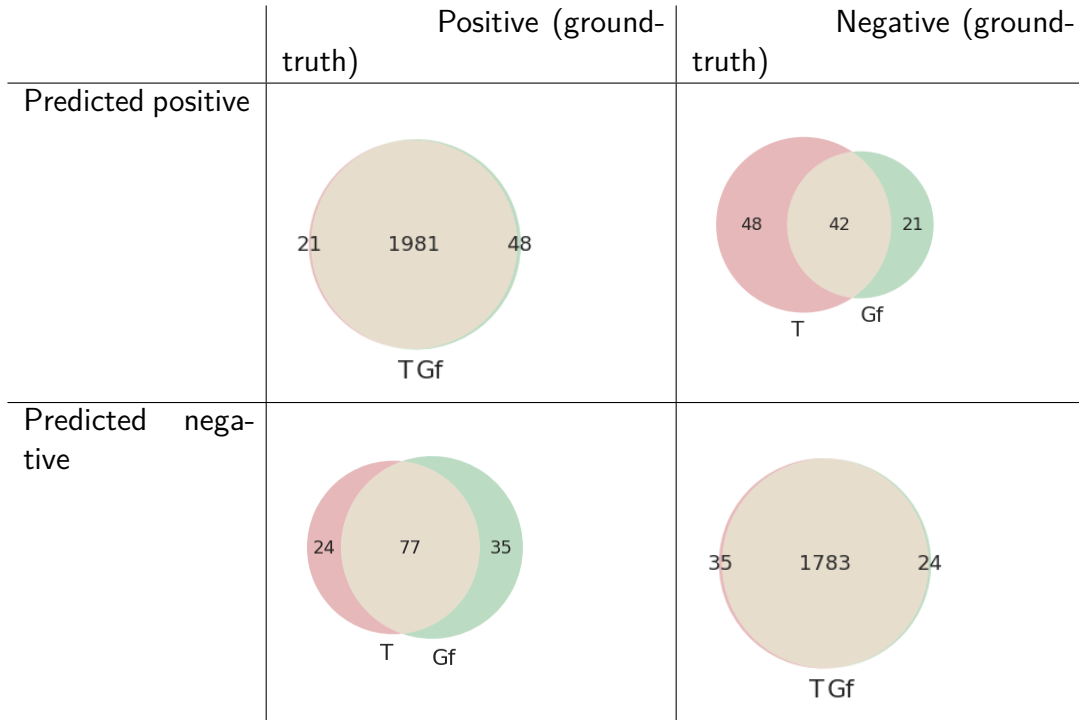


Figure 2.4 – Confusion matrix of the FlauBERT / \mathcal{T} (red) and FlauBERT / \mathcal{G}^f (green) models on the CLS-FR data. The Venn diagrams shows the proportions of shared examples for each category.

multiclass strategy *one-vs.-rest*. The number of iterations is set to a high value (2500), which ensures convergence for each of our experiments. Results for the same scenarios as above are presented for the MediaEval, CLS-FR and AG_news tasks in Table 2.3. For this type of classifier, the interest of the generated data appears for both scenarios and on the two datasets. In the case of substitution, the classifiers are slightly better than those trained on the original data. This demonstrates the importance of having a larger amount of data to capture form variants in texts (synonyms, paraphrases...) that the bag-of-words representations cannot otherwise capture as easily as the pre-trained embedding-based representations of the BERT models. In the scenario where data is used as a complement, the performance increase is even more marked and thus gets closer to the neural baseline, while having the advantages of a classifier considered more interpretable.

Impact of the quality of the generated data It is interesting to examine what is the influence of the quality of the generated data (including post-filtering) on the results of the

model	MediaEval			CLS-FR			AG_news		
	micro-F1	macro-F1	MCC	micro-F1	macro-F1	MCC	micro-F1	macro-F1	MCC
LR / \mathcal{T}	72.68	56.35	42.22	84.77	84.70	69.48	69.32	69.32	62.24
LR / \mathcal{G}^f	74.00	59.18	44.39	87.16	87.14	74.27	83.82	83.82	78.59
LR / $\mathcal{T} + \mathcal{G}^f$	75.46	59.64	45.83	88.36	88.34	76.69	83.47	83.47	78.65

Table 2.3 – Performance (%) of the LR/bag-of-words approach on the MediaEval, CLS-FR and AG_news datasets according to our scenarios of usage of the artificially generated data after filtering: without, substitution, complement.

final classifier. To study this, we simulate filtering done with classifiers of varying quality (accuracy). This is done simply by replacing, for randomly drawn examples of \mathcal{G}^f , the predicted class (by the generator and by the filtering classifier) with a randomly drawn class. For instance, for a (randomly picked) text generated for the class 'a' (and classified as 'a' by the filtering classifier), we change its label to class 'b' (randomly picked among the classes but 'a'). The number of examples undergoing this treatment is computed so that the errors inserted make the accuracy of the dataset drops to 80%, 70%, etc (considering that the original accuracy of the filtered dataset is 100%). The effect of these errors in the generated examples on the final performance of the complement and substitution strategies are presented in Figure 2.5 (MediaEval data) with logistic regression as the final classifier. As can be seen in this figure, empirical results about the influence of filtering quality are unsurprising. In the substitution scenario, the final performance is strongly dependent on the quality of the filtering classifier; in this case, a performance level equivalent to the original dataset is achieved when the accuracy of the filter exceeds 70 %. In the case of the complement scenario, the gain is significant as soon as the filter has an accuracy higher than random.

2.4 Conclusion

We have explored the interest of text generation for three text classification tasks (news categorization, fake news detection in tweets and sentiment analysis on product reviews). For replicability purposes, the training scenarios for the MediaEval and CLS-FR datasets presented in this chapter are made publicly available.

In a scenario where the original language resource or training data cannot be distributed, we have shown that it is possible to generate artificial data for supervised learning purposes. For state-of-the-art classifiers based on Transformers, **using the generated data without additional precautions degrades the performance** (compared to the one achieved

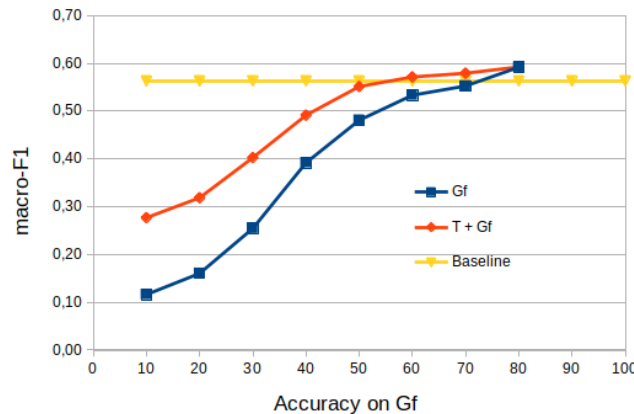


Figure 2.5 – Performance (macro-F1) according to the quality (accuracy in %) of the classifier filtering the artificially generated data; MediaEval dataset with logistic regression.

with the original data) **but in a contained proportion** (-4% accuracy). Yet, we have shown that **if the generated texts are automatically filtered with a classifier trained on the original data** (which can also be kept private), ensuring that texts belong to the desired classes, **it is possible to get equivalent or superior performance than with the original data**. The generation model and the filtering model can be kept private and the (filtered) generated texts can be distributed, which complies with use cases in which the sensitive original data cannot be distributed.

For classifiers exploiting bag-of-words representations, we notice in every case an improvement due to the larger amount of training data available, which makes it possible to get good results with more explainable ML methods if needed. In a scenario where artificial data is added to the original data, we have shown that classifiers benefit from additional data, including neural networks. This result is particularly positive for the bag-of-words approaches, which are more sensitive to reformulations, and which greatly benefit from the addition of these artificial examples. We have seen that for some datasets, it even allows the performance to get close to a BeRT-based model. We thus have **a good compromise between methods that are fast to train and more easily explainable, while having performance close to neural networks**. The explainability of misinformation detection models is crucial to detect potential biases, prevent harmful behaviors towards minorities and help human fact-checkers. We discuss this and introduce a novel explainability method in Chapter 7.

The best results are obtained when the generated data are filtered beforehand. A deeper study of the impact of the filtering and its quality are promising avenues.

In our experiments, the filtering was done automatically; manual correction of the data (of their classes) is also possible and may allow better results but with an additional annotation cost. Besides, it might be beneficial to **integrate the filtering step as a constraint during the generation of artificial data, using cooperative approaches introduced in Part II**.

Although these results are encouraging, the setup is different from the one we are interested in, in which one class is fully composed of generated data and the other of original data. We discuss this difference and the main issue that arises in the next chapter.

TEXTUAL GENERATIVE ADVERSARIAL NETWORKS

3.1 Detecting Generated Data

In the previous chapter, we demonstrated that data generated using language models can be used effectively in the training of discriminative models. However, this setting differs from our target setting by one important factor: the generated data are spread over every class of the model. Considering two classes $c \in \mathcal{C}$ for input news, *true* and *false*, the model is trained to output a probability vector for a sample x using cross-entropy loss:

$$\sum_{x \in \text{true}} \log p_D(\text{true} | x) + \sum_{x \in \text{false}} \log(1 - p_D(\text{true} | x)) \quad (3.1)$$

with $p_D(c | x)$ the discriminator probability of a sample x to belong to the class c .

When one class is composed of real human-generated data and the other is generated by a language model, the discriminative model can solve the task by detecting the source of the data: the classes of \mathcal{C} becomes *real* and *generated*. Hence, the model becomes a discriminator¹ trained to distinguish real data from generated data rather than learning

1. In the literature of generative adversarial networks, the term discriminator is used to refer to a classifier that can distinguish real from generated data. However, this distinction is not necessarily made in other domains such as constrained generation [180, 82] studied in the next part. Because our work arises from both domains, we chose to use discriminator as a synonym of classifier and make it clear when it refers to detecting generated data.

the intended task.

Detecting a neural generator is a more straightforward task for a model than considering the veracity of the information within a text, particularly when the generator is available. In a somewhat extreme manner, the generator can easily detect itself when used as a detector [405]. This is not surprising: given that the decoding schemes are based on its learned distribution, statistical methods leveraging it, such as simply counting the number of generated words that are among the most probable words for the generator, yield very high detection accuracy [117]. Additionally, asserting the likelihood of a sample under the model or using its hidden states enables zero-shot detection [405, 325]. Even without directly leveraging the model itself, models that have access to a sufficiently large set of generated texts achieve high detection accuracy. As discussed in the previous chapter, the generated samples contain artifacts that make them detectable, most notably the lack of syntactic and lexical diversity [117] or repetition [389, 144]. Moreover, the effective vocabulary of a language model is very small [151], and does not generate tokens that have a low probability, compared to human text that roughly follows Zipf's law [428]. Human texts are expected to be more diverse, use more unique words and fewer words from a "top-list", allowing blind-statistical methods (such as simple logistic regression on tf-idf) to succeed [111, 72].

Sampling in the full distribution reduces this phenomenon, but results in an increasingly high probability of picking a very unlikely token of the very long tail of the distribution that does not fit the sequence at all and creates gibberish sentences that are easily detectable [144]. This difference in the two distributions, induced by the decoding method, provides heavily discriminating clues for the detector, which will learn the "style" (effectively used tokens) by creating a language model of the generator that identifies it.

Finally, such model outputs tend to lack coherence [144, 46, 306], and sometimes jump from one topic to another totally unrelated one [23]. All these generation artifacts allow neural models (especially bidirectional Transformers such as RoBERTa [209]) to easily achieve very high detection rates, even if the discriminator is smaller than the generator (in terms of number of parameters) [405, 325, 25, 151, 78]. Detecting such artifacts is easier with longer sequences [303, 151, 325, 405] because repetitions and consistency errors are more likely to occur [144, 306], in addition to granting more observations for the model to perform the detection.

Since a model can easily detect generated texts when exposed to a sufficiently large amount of samples, it prevents the use of generated data as the only source of misinfor-

mation samples. However, this detection capability can be used as an insightful signal to improve the generator and create more realistic samples.

3.2 Reinforcement Learning and Sparse Rewards

3.2.1 Exposure Bias

Generative adversarial networks are a class of generative models that leverage a discriminator trained to distinguish real data from fake data to improve the quality of generated samples and overcome the limitations of traditional generative trainings. Indeed, textual generative models are usually trained with MLE, via teacher forcing [381] as introduced in Chapter 1. While appealing, this objective has several limitations. First, it is prone to overfitting because of a too strong exposure to the somehow limited ground-truth data considering the size of the space of valid sequences. Second, it is trained to optimize a unique GT, whereas many different sequences can represent the same semantic content and be considered as correct generations. Besides, the loss is defined at the token level, whereas the real interest lies in the finished sequences. This lack of sequence-level loss prevents the accurate optimization of sequence probabilities [379, 240], often resulting in degenerated texts (e.g., prone to repetition) [144]. Finally, and more importantly, MLE suffers from a mismatch between learning and simulation conditions, that is, the well-known exposure bias [274, 33]. During training, the model is only exposed to GT sequences, and never to its mistakes, while at test time (during inference), the model is conditioned on sequences of previously generated tokens which may have never been observed at training time and will suffer from error accumulation.

3.2.2 Reinforcement Learning

One way to overcome the limitations of TF is to directly optimize a sequence-level evaluation metric through Reinforcement Learning (RL) [398, 274]. This objective can be any standard NLP metric such as BLEU [426] or ROUGE [255]. These metrics are computed at the sequence-level by measuring the overlap ratio between n-grams of the sampled sequences and the GT references; BLEU is precision-oriented, while ROUGE is recall-oriented. Because of this, they are non-differentiable and are optimized using the REINFORCE algorithm [380]. REINFORCE estimates the gradient by sampling sequences from the model. The LM is then trained to optimize the log-likelihood of the best ones

by scaling the associated gradient based on the obtained reward (e.g., BLEU score). For a generator parameterized by θ , a generated sequence (Monte-Carlo sample) x and its reward $r(x)$, the gradient is: $\nabla_{\theta} L_{\theta}(x) = -r(x) \nabla_{\theta} \log p_{\theta}(x)$. A *baseline* b can be subtracted from the reward to reduce the variance of the gradient estimate, as long as it does not depend on the sample x (so the expected gradient is the same): $\nabla_{\theta} L_{\theta}(x) = -(r(x) - b) \nabla_{\theta} \log p_{\theta}(x)$.

3.2.3 Generative Adversarial Networks

However, the BLEU and ROUGE metrics are not totally correlated with human judgment, and optimizing them directly might lead to biased results rather than human-like ones [247]. A less biased metric is the score of a discriminator trained to differentiate the distributions of generated versus real texts. Guiding the generator toward a distribution that is indistinguishable from the real one would result in a perfect generator. Following this principle, in GANs [121] a discriminator network D is trained to distinguish real data from fake ones, the latter being generated using a generator network G trained to fool the discriminator. Both networks are trained as a min-max two-player game with value function $V(D, G)$, which is referred to as adversarial training:

$$\min_G \max_D V(D, G) = \min_G \max_D \left(\mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{x \sim p_{\theta}(x)} [\log(1 - D(x))] \right) \quad (3.2)$$

with p_{data} the probability distribution of the training dataset, p_{θ} the distribution of the generator and $D(x)$ the probability that x comes from the set of real data evaluated by the discriminator (i.e., $p_D(\text{real} \mid x)$).

Under some strong assumptions, [121] gives theoretical guarantees of convergence of the generator towards the distribution underlying observed training data, making generated samples undetectable. GANs allowed massive improvements in generative tasks in continuous domains such as image generation, thanks to their capacity to approximate continuous data distribution. However, for discrete data such as text, the gradient flow cannot be back-propagated from the discriminator to the generator, therefore the problem is commonly cast as a reinforcement learning problem using scores of the discriminator as rewards [398, 274]. These scores serve as a learning signal which is not affected by the exposure bias and does not rely on manually designed metrics that can be biased to evaluate the quality of a sample. Rather, the discriminator is used to discover a useful metric [274, 255, 301].

Note that, instead of RL-based approaches, some studies [186] leverage the Gumbel-

Softmax reparameterization trick [158, 218]. The issue with discrete GAN is that the discrete sampling operation that maps the output probability distribution from the language model to the discrete decision of the generated token is stochastic and discrete, preventing the backpropagation of the gradient flow. To solve the stochasticity issue, the reparameterization trick splits the stochastic sampling operation into a deterministic node coming from the network and a stochastic one that adds random noise, enabling the backpropagation into the deterministic node while letting the whole process stochastic. In practice, the noise g_i is drawn from a Gumbel distribution [327], providing a simple and efficient way to draw a sample x from the output categorical distribution of the network z by selecting the highest element from the perturbed distribution [219] ($x = \text{one_hot} \left(\arg \max_i [g_i + z_i] \right)$). Although this reparameterization allows backpropagating into the network while benefiting from the diversity of the stochastic process, the argmax operator still prevents the backpropagation because it is not differentiable. The softmax function is thus used as a continuously differentiable approximation of the argmax to generate a probability distribution over every element i of the vocabulary \mathcal{V} :

$$p_{\theta}(x_i \mid x_{1:t-1}) = \frac{\exp((z_i + g_i) / \tau)}{\sum_v \exp((z_v + g_v) / \tau)} \quad (3.3)$$

The resulting distribution becomes uniform as $\tau \rightarrow \infty$ and is identical to the original categorical distribution as $\tau \rightarrow 0$. The temperature parameter introduces a trade-off between discretiveness of the resulting distribution and variance in the gradients. Thus, the common practice is to start with an high τ and anneal it when the model becomes accurate to progressively get closer to the test setup.

3.2.4 Language Generative Adversarial Networks Falling Short

Still, GAN-based approaches suffer from both high variance and non-stationary reward distributions, leading to many instabilities, and therefore usually underperform models trained with traditional MLE [49, 307, 346, 224]. Learned distributions sacrifice diversity for quality to fool the discriminator [49], resulting in a worse trade-off than standard MLE. One of the main reasons behind the high variance of the gradient estimate in the RL-based approaches is the sparsity of the reward [418]. Indeed, among the combinatory space of tokens, only a few are legible and even fewer are good enough to fool the discriminator. It is way easier for the discriminator to learn to detect generated samples than it is for the generator to fool it. Hence, it is very hard for the generator to produce sequences that

obtain high rewards, which are the samples from which the model actually learns. To avoid a completely random search, the generator is first trained on available samples using MLE. The discriminator is then trained on the specific generator distribution, to maximize the alignment between its training and test distribution. Indeed, discriminators are shown to be strongly specialized for the current generator distribution (tied to the decoding method, temperature, training corpora, domain, etc.) [302, 16, 151, 25, 325]. Hence, because of this overspecialization, it is likely that at some point during the exploration, the generator will produce sequences that are out of the domain of the discriminator. These sequences will result in random rewards, and eventually make the generator collapse to a bad distribution if the random rewards are, out of luck, very large. This makes the training of textual GANs very unstable and requires to be very cautious in the sampling [302] and the learning rate. Even though a small number of generated samples are needed to adapt the detector [280], a few bad rewards yield during the adaption would make the generator collapse.

To solve the issue of sparse rewards, some works create denser rewards by using pairwise rewards, i.e., scoring a sequence relatively to another [418, 417]. However, comparative discriminators prevent the theoretic study of convergence behavior, as the convergence of non-Bernoulli GANs remains an open problem. One solution to help the generator to achieve high rewards is to guide it during the generation using the reward model itself, resulting in **a cooperative generation environment**.

PART II

COOPERATIVE GENERATION

INTRODUCTION TO CONSTRAINED GENERATION

Designing methods that embed the discriminator scores in the search necessarily leads to sequences with higher rewards on average. This will help the model to find correct sequences to learn from, improving its distribution, resulting in even better sequences at the next iteration, etc. This positive loop allows the generator to effectively learn to mimic the human distribution. A common analogy for GANs is the competition between a counterfeiter and a policeman mutually improving by observing the actions of each other. In the described setup, a crooked policeman, aware of the internal detection process, collaborates with the counterfeiter to produce the best possible counterfeit, enhancing the feedback loop. Yet, standard decoding methods only focus on the model likelihood, offering few options to have control on the generation process besides the prompt used to initiate the generation process. The goal of constrained textual generation is to find the sequence of tokens $x_{1:T}$ which maximizes $p(x_{1:T} \mid c)$, given a constraint c , for example being human-like for a discriminator and thus to generate less detectable samples. Although our target is this undetectability constraint, we started by exploring diverse constraints defined by more standard classifiers in Chapter 5 for the sake of comparison with state-of-the-art methods [304]. We return to this initial GAN objective in Chapter 6.

Even though a language model can be fine-tuned only on texts satisfying the constraint so it –hopefully– learns to model it, this approach is not scalable since it implies training multiple specific LMs (one per constraint). At the time of the experiments, tuning and

storing a LM for each constraint was very costly when even possible given the size of state-of-the-art LMs. The later introduction of low-rank adaptation of large language models (LoRA) [147] made it possible to tune and store different language models at a significantly lower cost by tuning the model using a low-rank approximation of the updates. In this manuscript, we consider the full fine-tuning of the language model, and we will discuss the relevance of the proposed methods in the light of LoRA in the conclusion of the next chapter. Few methods address the constrained textual generation without tuning one model for each constraint.

4.1 Class-Conditional Language Models

Class-conditional language models (CC-LMs), as the Conditional Transformer Language (CTRL) model [172], train or fine-tune the weights θ of a single neural model directly for controllable generation, by appending a control code at the beginning of a training sequence. The control code indicates the constraint to verify and is related to a class containing texts that satisfy the constraint. An illustration of CC-LM is given in Figure 4.1. For the sake of simplicity, we will denote without distinction the class, the constraint verified by its texts and the associated control code by c . Trained with different control codes, the model learns $p_\theta(x_{1:T} | c) = \prod_{t=1}^T p_\theta(x_t | x_{1:t-1}, c)$ for each of these control codes. The constraint can then be applied during generation by appending the corresponding control code to the prompt. While this method gives some kind of control over the generation, the control codes need to be defined upfront and the LM still needs to be trained specifically for each set of control codes. This is an important limitation since the current trend in text generation is the use of large pre-trained models which can hardly be fine-tuned without access to very large hardware resources (not considering LoRA) or even the weights themselves.

4.2 Cooperative Approaches

The general idea of cooperative generation [143, 303, 133, 61, 400, 84] is to guide a generative LM with an external discriminator D . Information from D is combined with the generator distribution during the generation to skew the generation towards samples that have a desired property defined by one class of the discriminator. The discriminator explicitly models the constraint by calculating the probability $p_D(c | x_{1:T})$ of the constraint

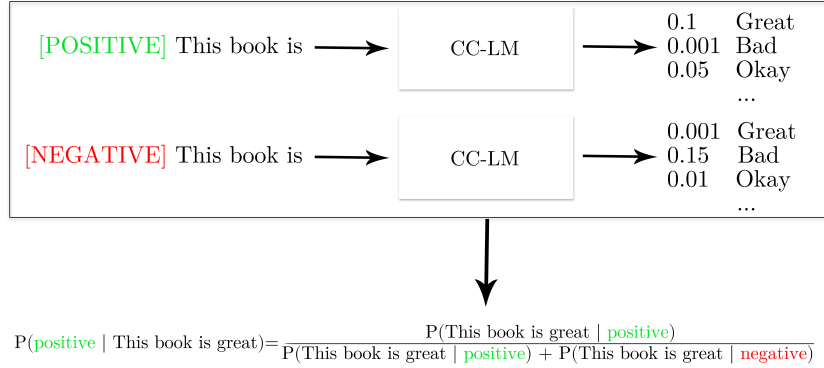


Figure 4.1 – Illustration of generative discriminators. Class-conditional language models compute the probability of the next token conditioned on a class using a control code. The classification probability is then computed using Bayes’ rule.

c to be satisfied by the sequence $x_{1:T}$. This probability is directly related to $p(x_{1:T} \mid c)$ through Bayes’ rule : $p(x_{1:T} \mid c) \propto p_D(c \mid x_{1:T})p_\theta(x_{1:T})$. We thus want to maximize both the likelihood of the language model and the discriminator probability. The discriminators have been used in different ways to explore the search space. In Plug And Play Language Model (PPLM) [82], the discriminator is used to shift the hidden states of a pre-trained Transformer-based LM towards the desired class at every generation step. PPLM can be used on any LM and with any discriminator. [61] uses the discriminator to filter out tokens that create a distribution discrepancy. In the work of [143, 303, 133], the space is first searched using beam search to generate a pool of proposals with a high likelihood $p_\theta(x_{1:T})$, and then the discriminator is used to re-rank them. However, beam search selects sequences with high likelihood regardless of the constraint, introducing a bias in the sampling. Indeed, the best overall sequence might have only average likelihood while satisfying the constraint perfectly.

Hence, it might be more suitable to take the discriminator probability into account during decoding rather than after generating a whole sequence. In this case, the discriminator is used at each generation step t to get the probability $p_D(c \mid x_{1:t})$ for each token of the vocabulary \mathcal{V} , and merge it to the likelihood $p_\theta(x_{1:t})$ to choose which token to emit. This is intractable given the usual large size of \mathcal{V} (in the order of ten thousand). In order to reduce the cost of using a discriminator on every possible continuation, [180] proposes to use CC-LMs as Generative Discriminators (GeDi). The method relies on the fact that the CC-LM computes $p_\theta(x_t \mid x_{1:t-1}, c)$ for all tokens of the vocabulary which can be used

to get $p_{\theta}(c \mid x_{1:t})$ for all tokens using Bayes' equation, as illustrated in Figure 4.1. This approach is thus at the intersection of tuning the LM and using a discriminator: it tunes a small LM (the CC-LM) to guide a bigger one. The major drawback of this approach is that CC-LMs are classifiers that perform worse than their discriminative counterparts. Thus, the authors propose an additional loss term to improve the classification power of these models, introducing a trade-off in the LM objective and the classification one.

Cooperative methods alleviate the training cost problem, as discriminators are easier to train than a LM. Moreover, any additional constraint can be defined a posteriori without tuning the LM, only by training another discriminator. A common drawback of all these approaches is their **lack of a long-term vision of the generation**. Indeed, the discriminator probabilities become necessarily more meaningful as the sequence grows and might only be trustable to guide the search when the sequence is (nearly) finished. When used in a myopic decoding strategy, classification errors will cause the generation process to deviate further and further. We thus propose to use a long-term planning strategy based on the Monte Carlo Tree Search to guide the generation process.

PPL-MCTS: CONSTRAINED TEXTUAL GENERATION THROUGH DISCRIMINATOR-GUIDED MCTS DECODING

Contents

5.1	Introduction	59
5.2	PPL-MCTS Method	60
5.3	Experiments	64
5.3.1	Experimental Setting	64
5.3.2	Automatic Metrics Results	68
5.3.3	Examples of Generation	69
5.3.4	Hyperparameters Exploration	70
5.3.5	Human Evaluation	72
5.4	Conclusion	73

In this chapter, we introduce a new cooperative decoding strategy to apply a constraint defined by an external classifier (e.g., being non-toxic, conveying certain emotions, using a specific writing style, etc.) to the generated text, without tuning the generative language model. It leverages the Monte Carlo Tree Search algorithm [77] (MCTS) to offer a long-term

view of the generation process and to efficiently explore the search space. We show that this approach outperforms state-of-the-art methods on three datasets and two languages. We also propose simpler methods based on re-ranking to fulfill this goal.

5.1 Introduction

Being able to add some constraints on the generated texts is useful for various situations. For example, it allows to create texts that follow a certain writing style, convey a certain emotion or polarity or to ensure that a generated summary contains correct information. More critically, it can be used to prevent the inherent toxicity of language models trained on the internet, or to not reproduce gender or race stereotypes.

As introduced in the previous chapter, besides the prompt used to initiate the generation process, there are few options to have control on the generation process. Most constraint generation methods necessitate to fine-tune the LM, which was very costly before the introduction of LoRA, when even possible given the size of state-of-the-art LMs. In this chapter, we propose new approaches to add such additional constraints on the texts but at decoding time through cooperative generation. We exploit a discriminator that is trained to determine if a text follows a given constraint or not; its output provides information to guide the generation toward texts that satisfy this expected constraint. In order to make the most of the discriminator information, we propose an original method based on the MCTS algorithm [77], namely **Plug and Play Language - Monte Carlo Tree Search (PPL-MCTS)**. We also propose simpler methods based on re-ranking to fulfill this goal. Both approaches do not require to fine-tune the LM; adding a new constraint can thus simply be done by providing a discriminator verifying if a text complies with what is expected.

Some cooperative approaches previously introduced use the discriminator at each generation step to find a sequence that satisfies the constraint. While this is less biased towards likelihood than re-ranking, they still lack a long-term vision of the generation. However, the meaning of a word can often only be defined once the context is fully known, that is, when the sequence is complete. Trying to optimize a score defined in the long horizon by making short-term decisions is very similar to common game setups such as chess, where the MCTS has proven to be really effective [316], which motivated our approach.

More precisely, our main contributions are the following ones:

1. we propose to use MCTS as a decoding strategy to implement constrained generation and we show, on 3 datasets and 2 languages, that it yields state-of-the-art results while offering more flexibility;
2. we also explore simpler generation methods based on re-ranking and show that this kind of approach, with low computational costs, can also be competitive if the diversity within propositions to re-rank is encouraged;
3. we provide a fully functional code implementing a [batched textual MCTS](#) working with the popular HuggingFace’s Transformers library [\[382\]](#)

5.2 PPL-MCTS Method

The approach that we propose is in line with methods using a discriminator to guide a large LM decoding, without the need to re-train it. Also, it can be applied to any LM with any discriminator, following the plug and play paradigm. Unlike previous approaches, it is able to have a long-term vision on what is generated. Being able to make a short-term decision (choice of the next token x_t at time step t) that is promising in the long run is based on the exploration of the search space. We propose here to use the Monte Carlo Tree Search for an efficient exploration of this space.

MCTS is very well suited for this problem for three reasons. First, it allows to get a local score (i.e., a score for the next token to emit) using finished sequences. Hence, this score is more meaningful than scores based only on the next step. Second, it allows to explicitly define the compromise between the exploitation of promising sequences (with a high likelihood), and the exploration of other potentially promising sequences (to not miss better sequences with a lower likelihood). The fact that regret, i.e the number of simulations done on a sub-optimal sequence, has a theoretical upper bound in MCTS [\[282\]](#) is a nice guarantee that the computation time is not wasted and the search is efficient. Finally, it outputs a solution at each iteration and so can fit our computational budget by allowing to adjust the quality of the solution to the calculation spent.

Text generation as tree exploration process. The search space of the text generation corresponds to a tree: its root is the prompt and the child of a node is its father’s sequence with one of the $|\mathcal{V}|$ possible tokens appended. In the case of constrained generation, the goal is thus to find the path, and therefore the sequence x , with the highest $p(x \mid c)$ possible without exploring the whole tree in width and depth. As mentioned previously,

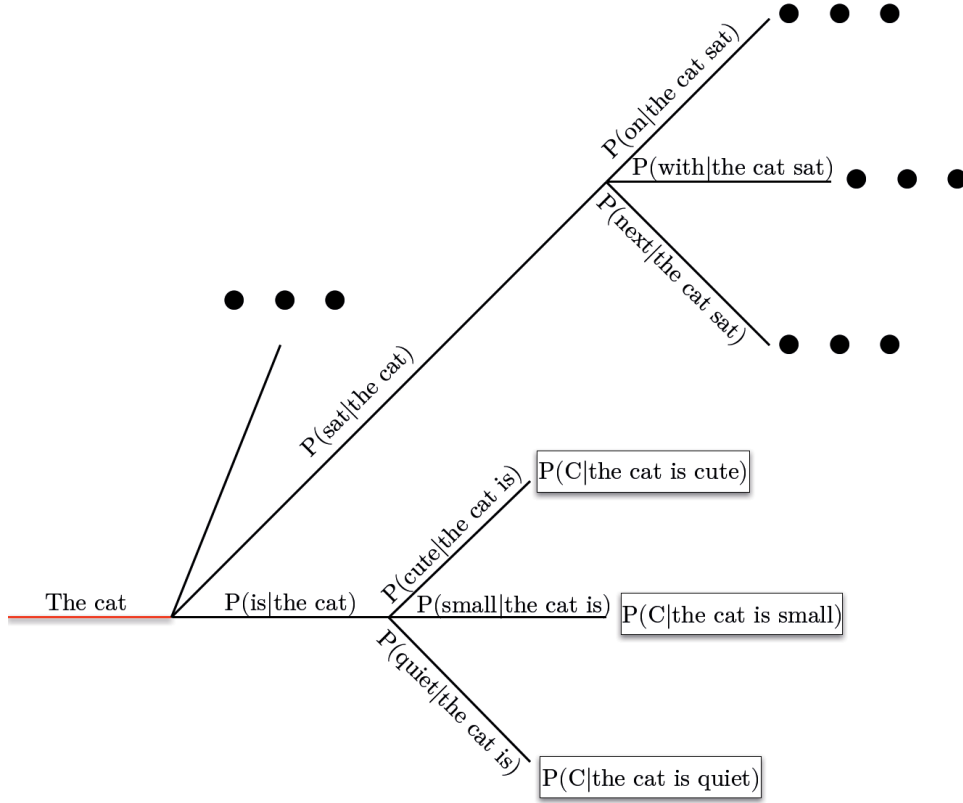


Figure 5.1 – Illustration of the constrained generation process as a tree exploration from the prompt **The cat**. Classification probabilities are only represented on completed sequences.

this probability can be computed as the product of the likelihood $p_\theta(x)$ and the probability given by a discriminator $p_D(c \mid x)$. An illustration of such a tree can be found in Figure 5.1, where the likelihood of x is forged by multiplying corresponding conditional probabilities along the path, and the classification probability is calculated at the terminal node.

Applying MCTS to text generation. MCTS is a heuristic-based iterative algorithm that uses randomness to solve deterministic problems that cannot be solved using traditional approaches, often because the search space is too large to be entirely explored. Each iteration consists of four consecutive steps. In the particular context of applying MCTS to text generation, we made some adaptations:

1. **Selection** Recursively choose children from the root to a node that has not been expanded yet. To only explore viable sequences, the probability $p_\theta(x_i \mid x_{1:t-1})$ of a given token x_i given by the LM is used during the selection phase. To this end,

the children chosen are those maximizing the Polynomial Upper Confidence Trees (PUCT) [282] as defined in [317]:

$$\begin{aligned}
 & \text{Selection score of a node } i \quad \text{Aggregated score of the node} \\
 & \text{Trade-off constant} \quad \text{Prior} \quad \text{Number of parent plays} \\
 & PUCT(i) = \frac{s_i}{n_i} + c_{puct} p_{\theta}(x_i | x_{1:t-1}) \frac{\sqrt{N_i}}{n_i + 1} \quad (5.1) \\
 & \underbrace{\frac{s_i}{n_i}}_{\text{Exploitation}} \quad \underbrace{c_{puct} p_{\theta}(x_i | x_{1:t-1}) \frac{\sqrt{N_i}}{n_i + 1}}_{\text{Exploration}} \\
 & \text{Number of node plays}
 \end{aligned}$$

with s_i the aggregated score of the node i , n_i the number of simulations played after this node, N_i the number of simulations played after its parent, and c_{puct} a constant defining the compromise between exploration and exploitation. In the task of constrained generation, we define the score of a sequence as its probability knowing the class $p(x | c)$.

2. **Expansion** If the selected node is not terminal, use the LM to expand it by creating its children.
3. **Simulation (roll-out)** Sample one of these children according to $p_{\theta}(x_i | x_{1:t-1})$, and go to a terminal node through a random walk or another pattern.
4. **Backpropagation** Aggregate the final score obtained at the terminal node ($p(x | c)$) to each parent until root. There are different strategies to aggregate scores, as computing the average between the actual score and the one being backpropagated, or taking the maximum of the two. We take the aggregated score s_i associated to the node i as the averaged probability over all simulations played after this node.

When the number of iterations has reached the allocated budget, the building of the tree stops. The token x_i selected for the current decoding step can be selected as the most played node amongst the root's children nodes, or the one with the highest aggregated score. We chose the most played one.

These adaptations of MCTS to constrained generation are summarized in Figure 5.2. Note that any language model can be used for defining the probability $p_{\theta}(x_i | x_{1:t-1})$ and any discriminator for scoring sequences, hence the name of our approach: Plug and Play Language - Monte Carlo Tree Search (PPL-MCTS). Compared to PPLM [82], our approach only requires the output logits while PPLM needs to access the LM to modify

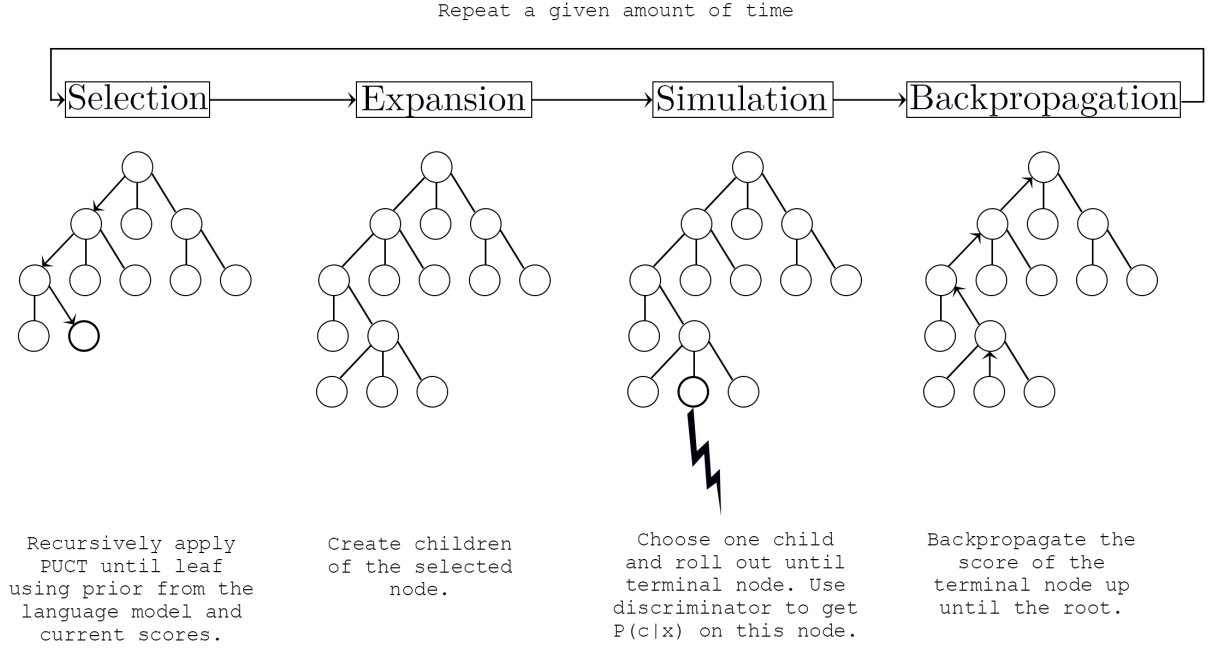


Figure 5.2 – MCTS application to text generation.

its hidden states. As some LM can only be used through access to logits (e.g., [GPT-3 API](#)), this makes our approach more plug and play than PPLM.

Model improvements. In order to allow finer control of how the constraint is applied, we introduce a parameter $\alpha \in [0, 1]$ to control the compromise between likelihood and constraint strength, modifying Bayes’ equation: $p(x | c) \propto p_D(c | x)^\alpha p_\theta(x)^{1-\alpha}$. Note that PUCT (5.1) already considers the likelihood of the sequence, favoring the selection of nodes with high likelihoods. Hence, even sequences generated with $\alpha = 1$ are correctly written. Setting $\alpha < 1$ forces the algorithm to explore solutions even closer to the language model. In our experiments, we set $\alpha = 1$ to strengthen the constraint, but we explore the trade-off between writing quality and constraint strength in Section 5.3.4.

To avoid expensive roll-outs, one may also assign a value to unfinished sequences at the cost of a less precise evaluation that may be not as meaningful as when doing roll-outs. Indeed, the discriminator can be trained on sequences with variable numbers of tokens, allowing it to be used at each node without the need for simulations. In this setup, the MCTS is used as an efficient compromise between exploration and exploitation, losing part of its long-view property but allowing to skew the exploration toward promising solutions.

Finally, during our first experiments, we observed that PPL-MCTS leads to repetitive patterns. This is very similar to what happens with greedy search, where a single sequence with a high likelihood is dominating the search. If such sequences also have pretty high discriminator scores, they will be repeated often. CTRL [172] offers a simple yet very powerful method to avoid noisy repetitions. It applies a scalar factor $I(i)$ to the temperature parameter τ of a given token x_i that penalizes this token if it is already in the input sequence. The probability of a given token becomes:

$$p'_\theta(x_i \mid x_{1:t-1}) = \frac{\exp(z_i/(\tau \cdot I(i)))}{\sum_v \exp(z_v/(\tau \cdot I(v)))} \quad (5.2)$$

with the *repetition penalty* $I(i) > 1$ if x_i is already in the prompt and 1 otherwise, and z the neural LM predicted logits over the vocabulary \mathcal{V} . Thus, probabilities of already emitted tokens are penalized, but if the language model gives a really high score to one token (hence, it is very confident that this *should* be the token to emit), it may still be selected as the output token.

5.3 Experiments

5.3.1 Experimental Setting

Performance Assessment The goal of constrained generation is to generate samples that 1) belong to a specific class while 2) keeping the language quality of the original LM, and 3) with enough diversity across samples. We chose three different metrics to evaluate each of these aspects: 1) accuracy, which is verified by an external "oracle" discriminator trained on a dataset disjoint from the one used to guide the generation; 2) perplexity, that is the average negative log-likelihood of the samples tokens, computed under an "oracle" LM, i.e an unconstrained LM trained on different data than the one used to train the constrained generator; 3) Self-BLEU score [426], which is the BLEU score [254] of a sample using the other samples as references: a high Self-BLEU score means that there is a lot of overlap between generated samples, and thus that the diversity is low. Such automatic metrics have known limitations [49] but results of human evaluation on the CLS dataset, detailed in Section 5.3.5, confirm that they provide a good overview of the performance.

Datasets Three different datasets are used in the experiments presented hereafter: [amazon_polarity](#) [410], CLS (from the [FLUE](#) [189] dataset) and [emotion](#) [297]. The first two

are Amazon reviews which have been labeled as positive or negative, so the intended task is to study the possibility of applying polarity to the generation. As CLS is in French, these two datasets will serve to ensure that the methods have the same behavior for different languages. An example of generated samples using PPL-MCTS for both classes of `amazon_polarity` is given in Figure 5.3. Emotion is a collection of tweets classified under eight basic emotions: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. This dataset is supposed to be more challenging since there are more classes and texts are smaller (only composed of one sentence), hence the model needs to precisely generate the target emotion with few tokens. It is worth noting that the 3 datasets have different sizes: 4,000,000 instances in total for `amazon_polarity`, 20,000 for emotion and 6,000 for CLS.

Data splits In practice, the studied dataset is split into two parts, each part being subdivided into train/val/test sets. The first train and validation sets are used to train and control the training of models used for the generation: the guiding classifier, the "vanilla" LM and the CC-LM. The test set serves to control their performance. The second ones are used to train the oracles which serve to compute the automatic evaluation metrics (the LM oracle and the classifier used to measure the accuracy). The test set allows to verify that these models are trustworthy for accurate evaluation. The test set of this second part is also used to forge prompts for the generation. We adapted the way we split each dataset depending on the original dataset splits. `Amazon_polarity` is composed of a training set of 3 600 000 examples and a test set of 400 000. We split both into two parts and kept 20% of each training set for validation. Emotion already comes with a train, test and validation set, hence we just split each into two parts. Finally, CLS is composed of a train set and a test set of 6000 examples. We split the training set in two and split the test set twice so we got two validation and test sets.

Each metric is evaluated on a pool of 900 generated samples using prompts never seen by models during training. We also adapted the prompts used to start the generation for each dataset depending on the data format. `Amazon_polarity` comes with a "title" column which corresponds to the title the user gave to the review. This field is directly used as the prompt. For the two other datasets, the prompts are the very first tokens of the text field. Because texts from emotion and CLS have different lengths, the sizes of prompts are also different: it is arbitrarily set to 6 tokens for CLS and 4 for emotion.

Baselines Besides PPL-MCTS, we propose several baselines and simple techniques. Most studies on re-ranking create proposals using beam search and then re-rank them using the product of likelihood and discriminator probability, limiting the diversity in the proposals pool. We propose re-ranking with different variations, in the way sequences to re-rank are produced, and in the way the final sequence is chosen in an attempt to improve such approaches. Three methods are tested to generate propositions: beam search [85] (with a beam size of 3), nucleus (top-p) sampling [144] (with $p=0.9$), as well as beam sampling (as described in [49]). For the final choice, we also propose three different methods: *argmax*, where the sequence that has the highest $p(x|c)$ is chosen; *first true*, where propositions are sorted by descending likelihood and the first sequence that belongs to the correct class according to the guiding discriminator is chosen; and *sampling*, where the distribution of $p(x|c)$ for the propositions is normalized and the chosen sequence is sampled following this distribution. Similarly to PPL-MCTS, the likelihood part of $p(x|c)$ is omitted (i.e., $\alpha = 1$) since sequences in the pool of propositions already have an high likelihood.

It should be noted that in our setting, a generated sequence corresponds to a document (e.g., a whole review). This choice makes sense for our datasets, but re-ranking at a smaller level (after each sentence, after x tokens...) would also be possible and might produce different results.

Methods from the literature We compare our results with methods from the literature. In particular, we test CC-LMs trained on the target task, similarly to CTRL, using greedy search as well as sampling for decoding. We also propose an implementation of CC-LM trained with the classification loss initially proposed for the GeDi method [180] to the improve classification accuracy. These CC-LMs are further used to implement the state-of-the-art GeDi model. In the experiments reported below, we report results for GeDi models trained with and without classification loss. Finally, we report the results of PPLM. For a fair comparison, the same discriminator and LM are used for our PPL-MCTS approach, the re-ranking approaches (baselines), and PPLM.

Evaluation Details For each method, a number of tokens equal to the average length of sequences of the dataset are generated: 98 tokens for *amazon_polarity*, 23 for *emotion*, and 137 for *CLS*. Fixing the number of generated tokens ensures fair comparisons between the tested methods. Indeed, even though perplexity and Self-BLEU metrics are normalized by the length of the sequence, these metrics can tend to penalize a model producing longer


```

<|startoftext|> The Revenge of making a good Halloween film. [SEP]????? I think this movie is a waste
of time. It's not scary, it's just plain stupid. The only good thing about this film is the soundtrack.<|endoftext|>
<|startoftext|> The Revenge of making a good Halloween film. [SEP] ive seen this movie a few times
and i love it. the acting is great, the story line is good, and the special effects are awesome. if you like horror
movies then go see this one.<|endoftext|>

```

Figure 5.3 – Example of two constrained generations using PPL-MCTS, one on the negative class, one on the positive class, using the same prompt (in bold) from `amazon_polarity` and same language model.

sequences: such a model has more risk to deviate and repeat itself, which would results in higher values compared to a model producing shorter sequences.

To run all of these methods, three different models are needed: one discriminator, a "vanilla" LM used as the generator, and the CC-LM used in the CTRL and GeDi approaches. For the discriminator used to guide the generation, we rely on [BERT-base-cased](#) [88] for the English datasets and [FlauBERT-large-cased](#) [189] for CLS. As vanilla LM, we use GPT-2 small models, relying on OpenAI's [pre-trained model](#) for the English datasets and on [belgpt2](#) for the French one. Given the particular format of data on our experimental datasets, the vanilla LM is trained on raw training sequences in order to produce texts corresponding to the task (for instance, reviews). The CC-LM is simply a fine-tuned version of the vanilla LM with the control code appended.

We tested three values for the temperature parameter for each proposed method (1.0, 1.1 and 1.2). For PPL-MCTS, we also studied the impact of c_{puct} by testing values 1.0, 3.0, 5.0 and 8.0 along with the different temperature values mentioned. We only report the results for parameters yielding the best accuracy score in the main body but every result can be found in Appendix A.1. The repetition penalty has been set to 1.2 as defined in CTRL. The number of MCTS iteration per token is set to 50, as well as the number of propositions for re-ranking, except for beam sampling where it is set to 10 because of memory limitations. We explored different numbers of iterations per token and found that 50 yields the best complexity/quality trade-off. Given the cost of roll-out for long sequences, we apply roll-out only on the emotion dataset to be able to run extensive experiments. As previously mentioned, without roll-out, MCTS loses a part of its long-view property but still allows to skew the exploration toward promising solutions. A study of the impact of the roll-out is detailed in a next sub-section. The parameters used for literature models are those provided by the authors. Experiments were conducted on a Quadro RTX 6000 with 80 GB of RAM.

5.3.2 Automatic Metrics Results

Results on the emotion, CLS and amazon_polarity datasets are reported in Table 5.1. The statistical significance against GeDi and PPLM is measured with a t-test with a significance level (p-value) of 1%. Results show that PPL-MCTS is competitive against task-specifically trained LMs on the constraint application aspect (high accuracy) while keeping a fair amount of diversity (low Self-BLEU) and staying close to the original distribution (low oracle perplexity). On all three datasets and metrics, it constantly yields top results; the only other method which is high-performing for all metrics and constant across the datasets is GeDi trained with the classification loss.

Another remarkable result is for the Sampling - Argmax method that selects among a pool of propositions generated using sampling, the one with the highest probability to be from the correct class. Thanks to the sampling used for generating propositions, its Self-BLEU is among the lowest of all reported values. Despite the simplicity and low computational cost of this approach, its accuracy is among the best on every dataset. These very good results should however be put into perspective of the very high perplexity of its generated texts. This indicates that the generated samples may be very different from those generated by a standard LM on this dataset as discussed in the next section.

Generation method	amazon_polarity			emotion			CLS		
	Accuracy ↑	5 - Self-BLEU ↓	Oracle perplexity ↓	Accuracy ↑	5 - Self-BLEU ↓	Oracle perplexity ↓	Accuracy ↑	5 - Self-BLEU ↓	Oracle perplexity ↓
Tuned LM									
CC-LM - Classloss	0.82	0.79	2.56 ^{*,†}	0.89 *	0.65 [†]	3.72 ^{*,†}	0.89*	0.04 ^{*,†}	50.6
CC-LM	0.91	0.71	3.21 [†]	0.52	0.13 ^{*,†}	11.1	0.66	0.06 ^{*,†}	31.5
GeDi - Classloss	0.96*	0.6*	5.16	0.88*	0.68	5.57*	0.94*	0.4	7.99*
GeDi	0.96*	0.6*	5.16	0.54	0.52 [†]	4.09 ^{*,†}	0.83*	0.31 [†]	11.9
Untuned LM									
PPLM	0.89	0.66	2.84 [†]	0.67	0.19 [†]	7.31	0.79	0.23 [†]	8.3
Beam - Argmax	0.88	0.85	3.14 [†]	0.72*	0.49 [†]	3.7 ^{*,†}	0.64	0.82	3.31 ^{*,†}
Beam - Sampling	0.86	0.84	3.27 [†]	0.7	0.46 [†]	3.69 ^{*,†}	0.6	0.82	3.37 ^{*,†}
Beam - First true	0.85	0.83	3.27 [†]	0.65	0.38 [†]	3.68 ^{*,†}	0.62	0.82	3.26 ^{*,†}
Beam sampling - Argmax	0.97*	0.73	3.82 [†]	0.67	0.48 [†]	3.88 ^{*,†}	0.88*	0.67	3.91 ^{*,†}
Beam sampling - Sampling	0.92	0.76	3.68 [†]	0.66	0.48 [†]	3.88 ^{*,†}	0.76	0.63	4.07 ^{*,†}
Beam sampling - First true	0.9	0.73	3.84 [†]	0.66	0.49 [†]	3.85 ^{*,†}	0.85*	0.71	3.8 ^{*,†}
Sampling - Argmax	0.99 ^{*,†}	0.17 ^{*,†}	16.5	0.87*	0.13 ^{*,†}	11.7	0.92*	0.12 ^{*,†}	14.3
Sampling - First true	0.89	0.07 ^{*,†}	85.9	0.82*	0.13 ^{*,†}	10.4	0.87*	0.14 ^{*,†}	13
Sampling - Sampling	0.88	0.17 ^{*,†}	16.3	0.81*	0.13 ^{*,†}	10.4	0.81	0.06 ^{*,†}	31.8
PPL-MCTS	0.97*	0.63*	5.69	0.84*	0.37 [†]	4.82 ^{*,†}	0.89*	0.54	4.98 ^{*,†}
PPL-MCTS - 10 tokens roll-out							0.95 *	0.57	5.07 ^{*,†}

Table 5.1 – Performance of constrained generation methods; from left to right: amazon_polarity, emotion, CLS datasets. † (resp. *) indicates statistically significant improvement against GeDi-classloss (resp. PPLM).

- **PPL-MCTS**
`<|startoftext|> realistic account [SEP] ive read this book and i think it is a very good account of the life of a young man. i would recommend to anyone who likes to know about the lives of people in america, or any other country that has been around for centuries.<|endoftext|>`
- **PPLM**
`<|startoftext|> realistic account [SEP] I enjoyed this book. It was realistic and I enjoyed the way the author described the people and places. I would recommend this book to anyone who is interested in the history of the United States.<|endoftext|>`
- **GeDi**
`<|startoftext|> realistic account [SEP] This is a real book. It's not just about the war, but also about the people who fought it. The author has a very realistic view of what happened in Vietnam and how it affected our country. I would recommend this book to anyone interested in learning more about Vietnam or any other war that took place during that time period<|endoftext|>`
- **Sampling - Argmax**
`<|startoftext|> realistic account [SEP] ive seen many of these. they mostly in the 20's, 30's and 40's and the luscious ones i know - men like this - there's an old saying that farts, u makin' u sell it..this movie has a lot of realism to it too! and i was totally impressed on how good the kids and the predator was! will it be hard for them to make more like this? i think it will! i read that war is going to be much<|endoftext|>`

Figure 5.4 – Examples of constrained generation using PPL-MCTS, PPLM, GeDi and Sampling - Argmax methods (from top to bottom) on the positive class of amazon_polarity, using the same prompt (in bold).

5.3.3 Examples of Generation

We provide examples of generation for amazon_polarity and emotion datasets using PPL-MCTS, PPLM, GeDi and Sampling - Argmax methods, respectively in Figures 5.4 and 5.5. Note that emotion texts are only one sentence while those of amazon_polarity are complete reviews. This difference motivated the choice of these datasets. Also, we preferred amazon_polarity over IMDb used in the GeDi and PPLM papers because of its bigger size, suitable format and because a French equivalent is available (CLS), which allows us to test another language with a similar dataset. As suggested by the reported high perplexity results, texts generated using Sampling - Argmax are rather different from the other samples and are badly written. Hence, exploring the accuracy/perplexity trade-offs achievable is interesting, notably through the α parameter.

```

— PPL-MCTS
<|startoftext|> i feel that working with a group of people who are so passionate about
the same thing is really important<|endoftext|>
— PPLM
<|startoftext|> i feel that working hard and caring for someone i don t care for is a
lot less selfish than i would be feeling for someone i<|endoftext|>
— GeDi
<|startoftext|> i feel that working with the ladies of the family is a wonderful thing
and i am very fond of the way they look and feel<|endoftext|>
— Sampling - Argmax
<|startoftext|> i feel that working at imgur for so many years is ill be devoted to
it<|endoftext|>

```

Figure 5.5 – Examples of constrained generation using PPL-MCTS, PPLM, GeDi and Sampling - Argmax methods (from top to bottom) on the 'love' class from 'emotion', using the same prompt (in bold).

5.3.4 Hyperparameters Exploration

Constraint strength through α As described in Section 5.2, a parameter α can be defined to control the relative importance of the discriminator score and the language model likelihood. Thus, this parameter allows to control the constraint application strength as it helps to define a trade-off between staying close to the original LM and satisfying the constraint. Note that in all of our experiments reported earlier, this parameter has been set to 1, focusing on the constraint application since the proposed methods already inherently provide legible texts. Here, as a proof of concept, we test a range of values for α , using the Sampling - Argmax method on the amazon_polarity dataset with the automatic metrics. We chose this method and dataset since it yields the best accuracy, but also exhibits a very high perplexity. In this case, it seems interesting to trade a bit of accuracy for better-written texts.

Results are roughly constant when α is lower than 0.98, so it has an impact only for values between 0.98 and 1. This is due to the fact that, for a long enough sequence, $p_\theta(x)$ is often relatively small compared to $p_D(c | x)$. This difference of scale annihilates the influence of α . This $[0.98; 1]$ interval thus corresponds to values of α that rescale $p_D(c | x)^\alpha$ and $p_\theta(x)^{1-\alpha}$ on a same order of magnitude. As shown in Figure 5.6, within this regime, we can observe a linear dependency between α and the accuracy as well as the perplexity. This illustrates that a trade-off can be obtained by tuning this parameter, allowing the definition of the strength of the constraint application which also defines how far the generation can be from the original LM distribution.

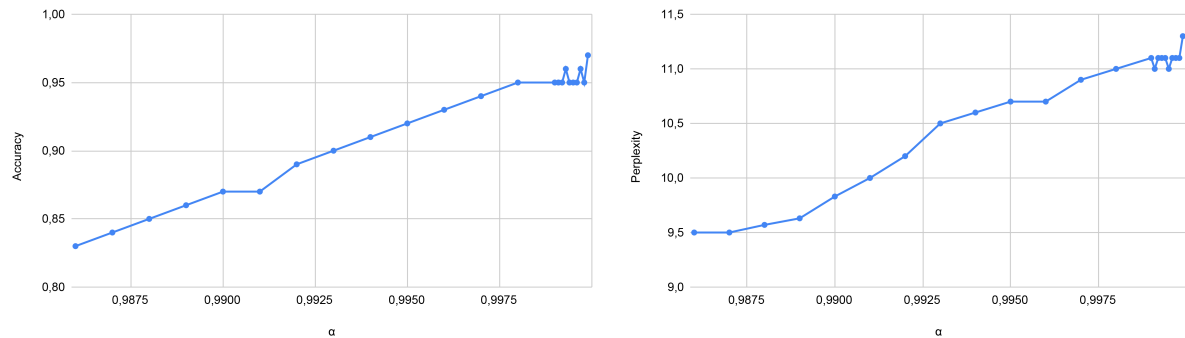


Figure 5.6 – Accuracy (left) and perplexity (right) of the Sampling - Argmax method according to the α parameter; amazon_polarity dataset

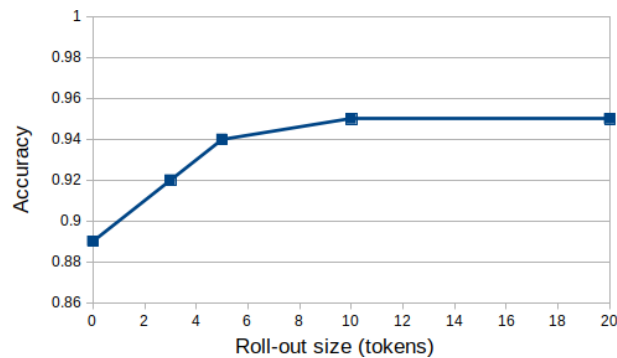


Figure 5.7 – Accuracy according to the roll-out size; CLS dataset

Effect of the roll-out Rolling out is costly for very long sequences, and the question of its usefulness necessarily arises. We study how rolling out for only a fixed number of tokens (instead of until the end of the sequence) influences the performance of PPL-MCTS. For this experiment, we use the CLS dataset and set the roll-out to 0 (original result), 3, 5, 10, and 20 tokens. As one can note in Figure 5.7, only 5 tokens allow PPL-MCTS to be on par with GeDi on this dataset. The roll-out size quickly improves accuracy, which then reaches a plateau. It suggests that having a horizon is really helpful but only up to a certain point. Besides, Self-BLEU and oracle perplexity values stay stable, varying respectively from 0.54 to 0.57, and from 4.98 to 5.18 as the roll-out size increases from 0 to 20. The roll-out size can thus be set accordingly to the compute budget, further defining the trade-off between cost and quality.

5.3.5 Human Evaluation

Since automatic metrics can be biased and may not faithfully represent the human judgment, we conduct a human evaluation to compare with the results obtained through oracles and confirm the results and the relevance of automatic metrics. Because of the annotation cost, we limit the tested methods to the two state-of-the-art methods (PPLM and GeDi), PPL-MCTS and the promising Sampling - Argmax. This allows to test if PPL-MCTS is indeed as efficient as GeDi and if both are better than the original PPLM. Also, this should confirm that the high perplexity of the Sampling - Argmax method is due to generated texts being very different from the ones generated by other methods. The evaluation has been performed on the CLS dataset by three volunteering colleagues, French native speakers. They labeled the same pool of reviews to measure the inter-annotator agreement.

The pool consists of 50 reviews (25 positive and 25 negative ones) randomly sampled for each method, which results in 200 reviews in total. Annotators were asked to go through this (randomly shuffled) pool and to give two scores for each review:

1. **Polarity.** Rate from 1 to 5 how well the text corresponds to the desired label (positive or negative). The text is rated 5 if it corresponds entirely to the expected label, down to 1 if it corresponds entirely to the opposite label. This score corresponds to the accuracy of the automatic metrics.
2. **Readability.** Rate from 1 to 5 how well the text is written. 5 corresponds to a text without any mistakes and which is perfectly understandable. The more mistakes or incoherence, the lower the score. This score corresponds to the perplexity of the automatic metrics.

The diversity within the pool of generated texts is complicated to evaluate and the Self-BLEU is fairly accurate as a diversity metric, so this property is not studied in the human evaluation.

We report scores averaged over the 3 annotators as well as the standard deviation in Table 5.2. A t-test against PPLM (GeDi being best on both scores) is applied to test statistical significance (with p-value=0.01). One can notice that the agreement between annotators is high and that the results are in line with conclusions from automatic metrics. GeDi, when trained with the classification loss, yields similar results as PPL-MCTS, in terms of constraint satisfaction and quality of writing. PPLM, on the other hand, generates samples of lower quality and has more difficulty applying the constraint. Finally, the

Generation method	Polarity	Readability
GeDi - Classloss	$4,46 \pm 0,08^*$	$4,19 \pm 0,28^*$
PPL-MCTS	$4,43 \pm 0,12^*$	$4,05 \pm 0,23^*$
PPLM	$3,74 \pm 0,08$	$3,12 \pm 0,19$
Sampling - Argmax	$4,00 \pm 0,11$	$2,83 \pm 0,33$

Table 5.2 – Results of the human evaluation on the CLS dataset (averaged over 3 annotators). * indicates statistically significant ($p \leq 1\%$) improvement against PPLM.

readability score of Sampling - Argmax confirms that it generates samples with a low quality. Its polarity score, while being higher than PPLM, is lower than expected: given the accuracy reported by the oracle, it should be close to GeDi and PPL-MCTS. It is most likely because evaluating the polarity of a badly written text is hard for a human, often resulting in the review being scored as neutral.

5.4 Conclusion

We show that it is possible to control the generation with the help of a discriminator that implements some expected constraints on the text during decoding. Our proposed methods, which mix the discriminator constraint and the generation, yield **performance that is equivalent to the best approaches based on LM full fine-tuning at lower training cost**. This flexible approach is very useful when using very large language models, whose full fine-tuning computational costs are prohibitive. In contrast, training a discriminator is easier and cheaper. On the other hand, such approaches have an additional cost during inference because of the cost of the discriminator being applied to candidate generations, motivating our study on this additional cost depending on the type of discriminator in Chapter 8. PPL-MCTS offers a solution for cases where training is too costly for the downstream application, where the language model is not directly accessible or to obtain better results at inference time with a fixed model.

As previously mentioned, LoRA [147], introduced after this study, reduces the cost of tuning and storing multiple language models. This is done by keeping the original base model fixed and only storing low-rank approximations of the updates of the different fine-tunings. This effectively enables the possibility to use different models trained

specifically for different constraints, thus lowering the benefits of cooperative approaches. However, they are still relevant in some aspects. While a model trained on texts satisfying a constraint might learn to model it **implicitly**, the application of the constraint is somewhat limited. This is in contrast with cooperative approaches that **explicitly** model the external constraint. This allows to ensure that the constraint is satisfied, while the implicit modeling of the LM does not prevent it from producing sequences that do not satisfy the constraint. It is easier to check if a sequence satisfies a constraint than to generate a sequence that does. This is one of the reasons why an external discriminator is helpful in GANs even though the generative model is trained only on human-written texts. Besides, by explicitly modeling the external constraint, **PPL-MCTS can not only be applied to any property that a discriminator can identify, but it can also work using other scoring methods** (human evaluation, factual consistency [400], logical constraints [212], unit tests for code generation [408], regular expressions, heuristic-based evaluation, ...) as long as the score reflects compliance with the expected property. This is not possible with LoRA. This makes the **use cases of PPL-MCTS numerous and diverse**. Besides the ones already mentioned, it can for example be used to add the required filtering part of Chapter 2 directly into the generation. In the remainder of this part, we explore some other applications such as its usage in the GAN setting in Chapter 6 and interpretability in Chapter 7. Finally, LoRA could be used jointly with cooperative approaches. Indeed, the same base model could be used for the generator and the discriminator, only differing by their LoRA weights. Thus, the cost in memory of the two models would be nearly the same as using only one [295].

Seeing text generation as a tree exploration process, GeDi indeed lowers the cost of width exploration but depth exploration is still an issue. Using GeDi for constrained generation is thus very similar to a standard maximum likelihood search which still lacks an optimal search method. On the other hand, Monte Carlo Tree Search provides an efficient way to explore the tree by determining the best local choice in the long run, lowering the cost of depth exploration. Thus, these two methods solve different facets of constrained generation, and the combination of the two appears as a promising perspective, which we explore in Chapter 8. Moreover, MCTS allows to **precisely define the best compromise between cost and quality** through the number of iterations and the roll-out size, while ensuring the efficiency of the search theoretically.

Finally, other research avenues are opened by this work. For methods yielding high perplexity, it would be interesting to explore how to set the α parameter in order to reach

the best compromise between accuracy and perplexity. Similarly, the size (number of tokens considered) of the roll-out in MCTS offers some ways to control the cost/performance compromise. An adaptive roll-out size, for example, rolling-out until the score of the discriminator is above or below a threshold as in [76], would seem particularly suited for texts. It should also be noted that fine-tuning a model and controlling the generation with a discriminator can be used jointly. For instance, one can use PPL-MCTS on a tuned LM, which will most likely result in even better performances because sequences considered during the search will have an overall higher quality for the considered task. To allow such explorations and reproducibility, our implementation is made [publicly available](#).

We have shown that the Monte Carlo Tree Search can be used to guide the generation in order to add the information from a classifier. In the context of textual GANs, sequences generated using the guidance from a discriminator should thus achieve higher rewards than sequences generated with solely the generator, which can help mitigate the rewards sparsity problem highlighted earlier in Section 3.2.

GENERATIVE COOPERATIVE NETWORKS FOR NATURAL LANGUAGE GENERATION

Contents

6.1	Introduction	77
6.2	Generative Cooperative Networks	78
6.3	Cooperating for Natural Language Generation	81
6.3.1	Learning Algorithm	81
6.3.2	Efficient Sampling	83
6.4	Experiments	86
6.4.1	Experimental Setting	86
6.4.2	Results and Discussion	88
6.5	Conclusion	91

In the previous chapter, we introduced a new cooperative generation method that leverages the Monte Carlo Tree Search to guide the generation and its performance on generating text satisfying a constraint defined by an external classifier. We experimented on constraining the generation towards a given class rather than human-like samples using a GAN discriminator for the sake of comparison with concurrent work on the subject [304].

This study explores our original idea to use texts cooperatively generated using the guidance of a GAN discriminator to train the language model and shows that it yields better generators. In this chapter, we further build upon this work and introduce a novel formulation of GANs for the discrete setting, which exhibits unpublished theoretical convergence guarantees. We then propose practical efficient NLG training algorithms relying on these theoretical results, based on various sampling schemes and corresponding re-weightings, resulting in our proposed Generative Cooperative Networks (GCN). Finally, we present state-of-the-art results for two important NLG tasks: Abstractive Summarization and Question Generation.

6.1 Introduction

As introduced in the previous chapters, the main idea of cooperative generation is that samples generated using the guidance of an external model will achieve higher scores for the guiding model. Considering a GAN discriminator as the external model means that the generated samples will be more realistic and human-like. Because discrimination is easier than generation, [303] shows that such guidance allows to generate better sequences. SelfGAN [304] builds upon this idea and trains the generator on the cooperatively generated samples using a standard MLE objective, in an expert-iteration learning scheme [14]. As cooperatively generated samples are better than samples generated using the LM alone, the LM will improve its distribution estimation. This will further improve cooperative samples for the next iteration, creating a positive loop. Such approaches overcome the rewards sparsity by guiding the generation using the signal from the reward model toward sequences that achieve high rewards. However, while such a kind of cooperative approach to produce accurate imitation learning samples is appealing, we argue that it turns out to be particularly unstable – even at the optimum. Indeed, as discussed in the next section, they can suffer from catastrophic forgetting and lose all the progress made up to this point in one update.

To address these shortcomings, we propose to take inspiration from [246] which introduced *Reward-augmented Maximum Likelihood* (RML), where samples to imitate are produced from a Boltzmann distribution $q(x) \propto \exp(f(x)/\tau)$, where $f(x)$ is an effectiveness metric of a sample x (e.g., the BLEU metric).

Adapting this framework with more flexible and learned quality metrics, like in GANs, approaches such as [304] employ, at each step t of the optimization process, a metric f

mainly depending on the current discriminator $D_t(x)$ for any sample x . We propose to rather consider $f(x) = \log(p_{t-1}(x)D_t(x))$ with p_{t-1} the previous generator distribution and D_t the discriminator at step t (trained on samples from p_{t-1}). This allows us to avoid instability issues and to present convergence guarantees under similar assumptions as those considered in the original GAN paper [121] for the continuous case. Then, we consider various efficient cooperative decoding approaches, which enable the practical optimization of such training processes, mainly based on Monte-Carlo techniques and importance sampling.

Our contribution is threefold:

- We propose a novel formulation of GANs for the discrete setting, which exhibits unpublished theoretical convergence guarantees;
- We propose practical efficient NLG training algorithms relying on these theoretical results, based on various sampling schemes and corresponding re-weightings;
- We present state-of-the-art results for two important NLG tasks: Abstractive Summarization and Question Generation.

Personal contribution Given the proximity of our studies with the LIP6, we decided to collaborate on the following studies during an internship that resulted in two publications, presented in this chapter and Chapter 8. In this work, which is an extension of our original idea to use cooperatively generated texts to train the generator, I actively took part in all the discussions - notably on the cooperative generation aspect - that led to the publication, as well as the redaction and the oral presentation.

6.2 Generative Cooperative Networks

In previous chapters, we used the decoder-only architecture for generative models that are given a prompt as input and that directly append generated tokens to it. We thus denoted both the prompt and the generated tokens by x . As introduced in Chapter 1, many NLG tasks (e.g., translation, summarization, question generation, etc.) imply a context as input associated with the target sequence. They are typically solved using an encoder-decoder architecture, with the context being processed by the encoder to further condition the output generated by the decoder. In order to emphasize this distinction, following the usual notations, we will denote by x the conditioning input and by y the corresponding conditioned sequence in this chapter.

Let $p_{\text{data}} : \mathcal{Y} \rightarrow [0; 1]$ be a target generative distribution, and assume we have access to training samples $y \sim p_{\text{data}}(y)$. The goal is to propose a training algorithm that computes a sequence of distributions $p_t(y)$ converging towards $p_{\text{data}}(y)$. In the following, we note $p_t : \mathcal{Y} \rightarrow [0; 1]$ a generator distribution obtained at iteration t of the algorithm, and $D_t : \mathcal{Y} \rightarrow [0; 1]$ a discriminator that outputs the likelihood for an outcome $y \in \mathcal{Y}$ of having been generated from p_{data} rather than from p_{t-1} .

Based on those definitions, a generic training process is given in Algorithm 1, where KL stands for the Kullback-Leibler divergence and h is a composition function that outputs a sampling distribution q_t depending on distributions given as its arguments. This training process unifies many different discrete GANs (e.g., SelfGAN [304] and ColdGAN [302]), as well as our present work, through the choice of function h applied to the current discriminator D_t and the previous generator p_{t-1} . Line 3 aims at finding the best possible discriminator D_t given distributions p_{data} and p_{t-1} , according to the classical objective to be maximized in GANs. Following the RML paradigm introduced by [246], line 4 seeks to optimize the generator distribution p_t by considering the minimization of the KL divergence $KL(q_t || p_t)$, according to a fixed behavior distribution q_t that includes feedback scores to be optimized (in our case, discriminator outputs). For cases where it is possible to efficiently sample from q_t , this is more efficient than considering a more classical reinforcement learning objective implying the reversed $KL(p_t || q_t)$, usually subject to high variance (e.g., via score function estimators).

Algorithm 1 *RML-GAN*

- 1: **Input:** a generator $p_0 \in \mathcal{G}$, a discriminator family \mathcal{D} .
 - 2: **for** iteration t from 1 to T **do**
 - 3: $D_t \leftarrow \arg \max_{D \in \mathcal{D}} \left(\mathbb{E}_{y \sim p_{\text{data}}(y)} [\log D(y)] + \mathbb{E}_{y \sim p_{t-1}(y)} [\log(1 - D(y))] \right)$
 - 4: $p_t \leftarrow \arg \min_{p \in \mathcal{G}} KL(q_t = h(p_{t-1}, D_t) || p)$
 - 5: **end for**
-

Let us first consider a setting where $q_t \triangleq h(p_{t-1}, D_t) \propto \exp(D_t)$, i.e., the sampling distribution only considers outputs from the discriminator. This corresponds to a direct application of the work from [246] for the GAN setting. For the sake of analysis, we consider the case where, at a given step t , the generator distribution is optimal, i.e., $p_t = p_{\text{data}}$ over the whole support \mathcal{Y} . In the next step $t+1$, the optimal D_{t+1} is equal to 0.5 for any sample from \mathcal{Y} . In this case, optimizing $KL(q_{t+1} || p_{t+1})$ with $q_{t+1} \propto \exp(D_{t+1})$ makes the generator

diverge from the optimum p_{data} , forgetting all information gathered until that point. This shows that the direct adaptation of GAN to discrete outputs is fundamentally unstable. Even though approaches such as SelfGAN use a pre-filtering based on the generator to allow tractable computation, this extreme setting illustrates instabilities that can occur with this family of recent state-of-the-art approaches. Discrimination cannot be all you need.

Therefore, we rather propose to consider a slightly different and yet much smoother optimization scheme, where both the generator and the discriminator *cooperate* to form the target distribution: $q_t \propto p_{t-1}D_t$. Such a choice for q_t allows us to prove the following theorem, which gives theoretical convergence guarantees for our collaborative training process (proof given in Appendix B.1).

Theorem 6.2.1. *With $q_t \propto p_{t-1}D_t$, if the generator and discriminator architectures have enough capacity, and if at each iteration of Algorithm 1 both optimization problems reach their respective optimum (i.e., $D_t(y) = \frac{p_{\text{data}}(y)}{p_{\text{data}}(y) + p_{t-1}(y)}$ for any $y \in \mathcal{Y}$ (line 3) and $KL(q_t \propto p_{t-1}D_t || p_t) = 0$ (line 4)), then, starting from p_0 such that $p_0(y) > 0$ whenever $p_{\text{data}}(y) > 0$, p_t converges in distribution to p_{data} when $t \rightarrow +\infty$.*

As for classic continuous GANs, the neural architectures used to define generator and discriminator function sets \mathcal{G} and \mathcal{D} in practice represent a limited family of distributions, depending on their depth and width. However, the given theorem allows us to expect reasonable behavior for sufficiently powerful architectures. The following theorem relaxes the constraint on the optimal discriminator (proof in Appendix B.2).

Theorem 6.2.2. *With $p_t \propto p_{t-1}D_t$, and if the discriminator is sufficiently trained, i.e., we have $\log \eta = \min \left(\mathbb{E}_{y \sim p_{\text{data}}(y)} [\log(D_t(y))], \mathbb{E}_{y \sim p_{t-1}(y)} [\log(1 - D_t(y))] \right)$, with $\eta \in]\frac{1}{2}; 1[$, then we have at each iteration of Algorithm 1: $\Delta_t \triangleq KL(p_{\text{data}} || p_t) - KL(p_{\text{data}} || p_{t-1}) \leq \log(\frac{1}{\eta} - 1) < 0$.*

In other words, it suffices that both parts of the discriminator objective exceed the random accuracy (i.e., $1/2$) in expectation to make $q_t \propto p_{t-1}D_t$ a useful target to be approximated at each step. Even with only a few gradient steps at each iteration, we can reasonably assume that the parameters space is smooth enough to guarantee the convergence of the algorithm, with almost only useful gradient steps. We also note that

the better discriminator is (i.e., higher η), the more useful a move from p_{t-1} to p_t is (in terms of KL).

Getting back to Algorithm 1, at line 4, optimization can be performed via gradient descent steps $\nabla_{p_t} KL(q_t||p_t)$, which can be rewritten via Importance Sampling (IS) as:

$$\nabla_{p_t} KL(q_t||p_t) = - \mathbb{E}_{y \sim q_t(y) \propto p_{t-1}(y) D_t(y)} [\nabla_{p_t} \log p_t(y)] \quad (6.1)$$

$$\begin{aligned} &= - \mathbb{E}_{y \sim p_{t-1}(y)} \left[\frac{q_t(y)}{p_{t-1}(y)} \nabla_{p_t} \log p_t(y) \right] \\ &= - \frac{1}{Z_t} \mathbb{E}_{y \sim p_{t-1}(y)} [D_t(y) \nabla_{p_t} \log p_t(y)] \end{aligned} \quad (6.2)$$

with $Z_t = \sum_{y \in \mathcal{Y}} p_{t-1}(y) D_t(y)$ the partition function of q_t . Note that, with the exception of the partition score Z_t that acts as a scale at each step, the considered gradient is closely similar to what is optimized in classic discrete GANs via reinforcement learning (i.e., policy gradient optimization of p_t and the discriminator score as the reward), when only one gradient update is performed at each iteration.

The effect of this scaling factor is highlighted when written as an expectation: $Z_t = \mathbb{E}_{y \sim p_{t-1}(y)} [D_t(y)]$. From this, it is clear that Z_t is maximized when the generator distribution coincides with D_t , i.e., when p_{t-1} allocates the best probability mass for samples judged as the most realistic by the current discriminator. In the absence of such a normalization term, classic GAN approaches need to set an arbitrary learning rate scheduling to avoid the explosion of gradient magnitude as p_t gets closer to p_{data} (near the optimum). Our approach, naturally stabilized by Z_t , does not require such difficult tuning to ensure convergence – as verified empirically in Section 6.4.

6.3 Cooperating for Natural Language Generation

In this section, we first present the extension of Algorithm 1 to the setting that implies a context as input before discussing its practical implementation as well as the sampling strategies that enable its efficient use in real-world settings.

6.3.1 Learning Algorithm

Let Γ be a training set of N samples (x^i, y^i) where each $x^i \in \mathcal{X}$ is a (possibly empty) context (assumed to be sampled from a hidden condition distribution p_x) and

$y^i \sim p_{\text{data}}(y^i|x^i)$ is the corresponding observation. Algorithm 2 gives the practical implementation of Algorithm 1 for large-scale NLG tasks. It considers parametric distributions p_θ and D_ϕ , implemented as deep neural networks¹, with respective parameters θ and ϕ . Thus, $p_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow [0; 1]$ is the generative conditional distribution, where $p_\theta(y|x) = \prod_{j=1}^{|y|} p_\theta(y_j|x, y_{0:j-1})$ with $p_\theta(y_j|x, y_{0:j-1})$ the categorical distribution for token j of sequence y over the vocabulary, given the context x and the sequence history $y_{0:j-1}$. Also, $D_\phi : \mathcal{X} \times \mathcal{Y} \rightarrow [0; 1]$ is the conditional discriminative distribution, where $D_\phi(x, y)$ returns the probability for sequence y of having been generated from p_{data} rather than p_θ given the context x .

The discriminator is trained at line 5 of Algorithm 2, on a batch of m samples of contexts, associated with corresponding sequences y from the training set and generated sequences \hat{y} from the current generator. Consistently with [303], to effectively drive the cooperative decoding process in guided sampling strategies \hat{q} (see below), the discriminator is trained, using a left-to-right mask, on every possible starting sub-sequence $y_{1:j}$ in each sample y (i.e., taken from its start token to its j -th token), with $j \leq l$ and l standing for the max length for any decoded sequence. As introduced in Chapter 5, this enables discriminator predictions for unfinished sequences, allowing to avoid the complex roll-outs in MCTS.

Line 7 of Algorithm 2 performs a gradient descent step for the generator, according to samples provided by a sampling strategy \hat{q} . Ideally, consistently with Equation 6.1, training samples should be provided by $q_{\theta,\phi}(y|x) \propto p_\theta(y|x)D_\phi(y|x)$. However, directly sampling from this distribution is intractable. Various sampling strategies can be considered, using a weighted importance sampling scheme to unbiased gradient estimators in line 7. For the task of unconditional generation (i.e., empty contexts x) and the case where $\hat{q} = p_\theta$, we can show that this is equivalent, up to a constant factor, to the gradient estimator given in Equation 6.2, with expectations estimated on the current batch, since in that case w^i reduces to $D_\phi(x^i, \hat{y}^i)$. However, more efficient sampling strategies \hat{q} can be employed, as discussed in the following. The process is illustrated by Figure 6.1, which compares the architectures of classical GANs with our GCN approach.

1. Transformer T5 [271] in our experiments

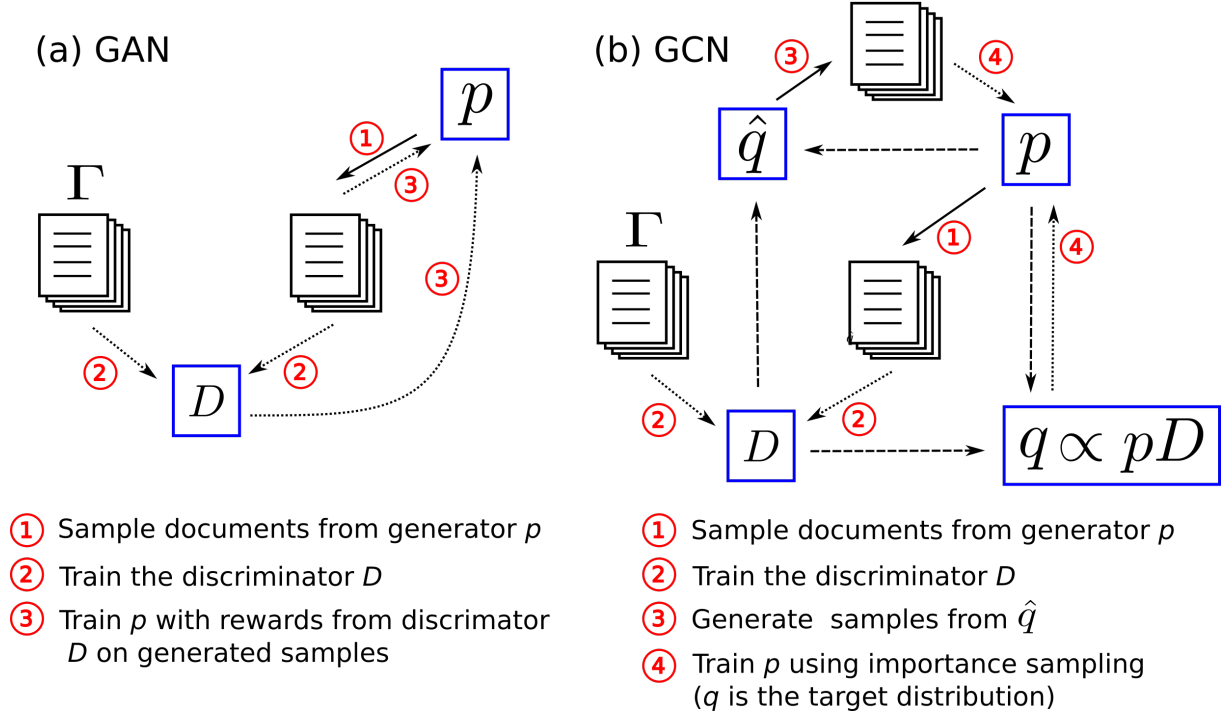


Figure 6.1 – Training GANs vs GCNs. Solid arrows stand for sampling, dashed arrows depict dependence between distributions and dotted ones denote training. (a) Classical discrete GAN where: 1) the generator p samples sequences; 2) the discriminator D is updated according to these generated sequences and those from the training set Γ ; 3) the scores given by the discriminator D are used as the reward in a policy gradient update of p . (b) Our GCN approach where \hat{q} and q respectively stand as the behavior and target distributions, which are defined by a cooperative scheme between D and p . After sampling from \hat{q} in 3), q is used in an importance sampling weight for the update of p in 4), which corresponds to the minimization of $KL(q||p)$. The distribution q is set as $q(x) \propto p(x)D(x)$ to ensure convergence, and \hat{q} can take various forms, the closer to q the lower the variance.

6.3.2 Efficient Sampling

To minimize the variance of gradient estimators, we need to sample sequences following a distribution as close as possible to $q_{\theta,\phi}(y|x) \propto p_{\theta}(y|x)D_{\phi}(x,y)$. While directly sampling from such a non-parametric distribution is difficult, and given that rejection-sampling or Markov Chain Monte Carlo (MCMC) methods are very likely to be particularly inefficient in the huge associated support domain, we build on cooperative generation to sample informative sequences, that are both likely for the generator p_{θ} , and realistic for the discriminator D_{ϕ} . Note that an alternative would have been to exploit the maximum entropy principle [427] to learn a neural sampling distribution \hat{q}_{γ} as

Algorithm 2 *Generative Cooperative Networks*

- 1: **Input:** generator p_θ with parameters θ , discriminator D_ϕ with parameters ϕ , training set Γ , sampling strategy \hat{q} , batch size m , max sequence length l .
 - 2: **for** $t = 1, \dots, T$ **do**
 - 3: Sample $\{(x^i, y^i)\}_{i=1}^m$ from Γ
 - 4: $\forall i \in \llbracket 1; m \rrbracket$: Sample $\hat{y}^i \sim p_\theta(\hat{y}^i | x^i)$;
 - 5: $\phi \leftarrow \phi + \epsilon_\phi \sum_{i=1}^m \sum_{j=1}^l \left[\nabla_\phi \log D_\phi(x^i, y_{0:j-1}^i) + \nabla_\phi \log(1 - D_\phi(x^i, \hat{y}_{0:j-1}^i)) \right]$
 - 6: $\forall i \in \llbracket 1; m \rrbracket$: Sample $\hat{y}^i \sim \hat{q}(\hat{y}^i | x^i)$;
 - 7: $\theta \leftarrow \theta + \epsilon_\theta \left[\frac{1}{\sum_{i=1}^m w^i} \sum_{i=1}^m w^i \nabla_\theta \log p_\theta(\hat{y}^i | x^i) \right]$ **with** $w^i = \frac{p_\theta(\hat{y}^i | x^i) D_\phi(x^i, \hat{y}^i)}{\hat{q}(\hat{y}^i | x^i)}$
 - 8: **end for**
-

$\arg \max_{\hat{q}_\gamma} \mathbb{E}_{y \sim \hat{q}_\gamma(y|x)} [\log p_\theta(y|x) + \log D_\phi(x, y)] + \mathcal{H}_{\hat{q}_\gamma(\cdot|x)}$, with \mathcal{H}_q the entropy of distribution q . This would however imply a difficult learning problem at each iteration of Algorithm 2, and a sampling distribution \hat{q}_γ that lags far behind $q_{\theta, \phi}$ if only few optimization steps are performed.

Sampling Mixtures

We consider the use of variance reduction techniques when sampling from the generator distribution, which can be long-tailed, thus leading to unreliable sequence samples. In particular, nucleus sampling [144] has been shown to produce higher quality texts than more classic sampling strategies, including beam search and low temperature-based sampling [302]. We denote in the following $p^{nucleus=\sigma}$ the truncation of distribution p on the minimal set of tokens of \mathcal{V} that have a cumulative probability higher or equal to σ .

Using this technique for defining \hat{q} in our Algorithm 2 could allow avoiding usual text degeneration issues [144], which would benefit to our generative learning process by providing better-formed sequences to the discriminator. However, importance sampling demands that $\hat{q}(y) > 0$ for any $y \in \mathcal{Y}$ such that $q(y) > 0$. A direct use of nucleus sampling as $\hat{q} = p^{nucleus=\sigma}$, or even more a classic beam search, cannot guarantee this property, which might involve ignoring many useful parts of \mathcal{Y} in the gradient estimation, hence implying biases.

To cope with this, we propose to follow ColdGAN [302], which considers sampling distributions \hat{q} as mixtures, ensuring that both properties, i.e., IS consistency and high-

quality samples, are verified. Formally, we use

$$\hat{q}_\theta(y|x) = \epsilon p_\theta(y|x) + (1 - \epsilon) p_\theta^{\text{nucleus}=\sigma}(y|x) \quad (6.3)$$

where ϵ stands for a small probability for sampling from the true generator distribution rather than using a nucleus decoding ($\epsilon = 0.1$ and $\sigma = 0.1$ in our experiments), thus ensuring the validity of our IS estimator. Please also note that, using such a mixture trick, each IS weight is upper-bounded by $D_\phi(y|x)/\epsilon$, which greatly limits gradient explosion issues usually associated with the use of IS in RL (or over-weighting of unlikely sequences in weighted IS).

Guided Sampling

Next, we propose to use cooperative decoding strategies to get a sampling distribution closer to $q_{\theta,\phi}$. More specifically, we propose to employ the Monte Carlo Tree Search strategy introduced in the previous chapter. Using left-to-right decoding strategies, it can happen that all sequence candidates are judged as unrealistic by the discriminator, avoiding any useful learning signal for the generator. MCTS allows to deal with this strong limitation of myopic decoding, by anticipating the final utility of the successive decisions. There are minor differences with the MCTS process presented in Chapter 5. First, roll-outs are replaced by evaluations on corresponding unfinished sequences. This reduces the variance of the MCTS sampling at the cost of learning a value network able to score incomplete sequences [190] that outputs the expectancy of a state. Also, to be consistent with SelfGAN, during back-propagation, the updated score is the maximal score between the back-propagated score and the actual score of the node.

Cooperative Learning with MCTS To use this MCTS process to guide the generator decoding toward sequences of high discriminator scores, in our learning Algorithm 2, we re-use the same mixture trick as for Nucleus Sampling discussed above:

$$\hat{q}_\theta(y|x) = \epsilon p_\theta(y|x) + (1 - \epsilon) p_\theta^{\text{mcts}}(y|x) \quad (6.4)$$

where $p_\theta^{\text{mcts}}(y|x)$ is a Dirac centered on the decoded sequence from the MCTS process in the conditional case (when contexts x are available), and the MCTS sampling distribution (according to number of visits, as described in the MCTS decoding process) in the unconditional case. Again, $\hat{q}_\theta(y|x) > 0$ whenever $y \in \mathcal{Y}$ such that $q_{\theta,\phi}(y|x) > 0$, and the

IS weights are upper-bounded by $D_\phi(y|x)/\epsilon$.

6.4 Experiments

6.4.1 Experimental Setting

To evaluate the framework, we experiment on standard complementary unconditional and conditional NLG tasks, with the following datasets:

Unconditional NLG – Following the same setup as in many related studies (e.g., [302, 49]), we first compare our approaches with NLG baselines on the task of unconditional text generation, where the aim is to reproduce a given unknown generative distribution of texts from samples, on the EMNLP2017 News dataset.

Question Generation – The task consists in generating the question corresponding to a given text and answer. For this task, we use the SQuAD dataset [272], composed of 100K triplets of Wikipedia paragraphs, factual questions, and their answers. Examples of questions generated using GCN are given in Figure 6.2.

Abstractive Summarization – The aim of this standard sequence-to-sequence task is to produce an abstract given an input text. We use the CNN/Daily Mail dataset (CNNDM) [239], composed of 300K news article/summaries pairs. Target summaries consist of multiple sentences, allowing us to evaluate models on longer texts than for the Question Generation task.

To compare the models, we consider the standard BLEU [254] and ROUGE [204] metrics. For the task of unconditional NLG, where diversity is of crucial importance (because there is no diversity within the context), we follow [49], who proposed to plot results as curves of BLEU (i.e., with samples classically compared to ground truth references, measuring accuracy) vs. Self-BLEU (i.e., with generated samples compared to themselves, measuring diversity). This is done by sampling texts for various temperature settings (i.e., temperature of the softmax on top of the generator).

We compare our models with the following baselines:

- **MLE** – We naturally consider as an important baseline the T5 model trained via teacher forcing. It is furthermore used as a starting point for all models (unless specified).
- **ColdGAN** – This model was one of the first GANs to outperform MLE for NLG

Input context: Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24 euros 10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals	
— Expected answer: Super Bowl	Generated question: How is called the American football game which determines the NFL champion?
— Expected answer: golden anniversary	Generated question: As this was the 50th Super Bowl, what was emphasized by the league?
— Expected answer: 50th Super Bowl	Generated question: The league emphasized the "golden anniversary" during what Super Bowl?

Figure 6.2 – Examples of questions generated using GCN. The questions to be generated are conditioned on the input context and the expected answer.

tasks [302]. Its main contribution was to introduce the use of a sampling strategy with lowered softmax temperature during training, with the objective of stabilizing the training process. We use its best-reported version, which considers a mixture with nucleus sampling.

- **SelfGAN** – The work presented in [304] uses an expert-iteration algorithm in combination with various cooperative decoding strategies. In the following, we report results from its version using a MCTS process, which recently obtained state-of-the-art results on the three considered NLG tasks.
- **GCN** – The Generative Cooperative Networks introduced in this chapter. Three versions of Algorithm 2 are considered in the experiments: $\text{GCN}^{\hat{q}=p}$, which corresponds to a classic GAN with implicit dynamic scheduler induced by partition $z_t = \sum_i w^i$, $\text{GCN}^{\hat{q}=\text{Nucleus}}$, which considers a mixture with Nucleus Sampling as defined in Equation 6.3, and $\text{GCN}^{\hat{q}=\text{MCTS}}$, which considers a mixture with a discriminator-guided MCTS, as defined by Equation 6.4.
- **GAN** – For ablation study purposes, we also consider similar versions of our implementation of Algorithm 2 but without the use of normalization, respectively called $\text{GAN}^{\hat{q}=p}$, $\text{GAN}^{\hat{q}=\text{Nucleus}}$ and $\text{GAN}^{\hat{q}=\text{MCTS}}$. The normalization is replaced by a linear learning rate scheduler tuned on a validation set for $\text{GAN}_{+\text{scheduler}}^{\hat{q}=p}$, $\text{GAN}_{+\text{scheduler}}^{\hat{q}=\text{Nucleus}}$ and $\text{GAN}_{+\text{scheduler}}^{\hat{q}=\text{MCTS}}$.

For each model, any decoding method could be applied at inference time, independently

of the training scheme. In the following, unless specified otherwise, we report results obtained with a classic beam search decoding (with a beam size of 3) for all the experiments.

Implementation Details The experimental settings are similar to those used in the SelfGAN paper [304]. In all the experiments, the models are initialized with the Seq2Seq T5 model from [271]. Unless specified otherwise, we use the T5-small architecture (60M parameters), as implemented in the HuggingFace library [382]. For the discriminators, we frame the classification task as a text2text task where the model has to generate either the token *human* or *machine*. This allows to use again T5-small for all experiments, removing possible bias from architecture differences between the generator and the discriminator. We start by training via teacher forcing a model corresponding to the MLE baseline. All GANs are initialized from this MLE model. During this pre-training, a learning rate fixed to $5e - 6$ is used for both the discriminator and the generator, and the number of epochs is set to 5.

We tested on a validation set different values for the hyperparameter c_{puct} ($\{1.0, 2.0, 3.0, 4.0\}$) and the number of simulations per token ($\{5, 10, 25, 50, 100\}$). Interestingly, we found similar results to those of PPL-MCTS. Indeed, $c_{puct} = 3.0$ gives the best results and we do not observe significant improvement for numbers of simulations above 50. We thus only report the results with $c_{puct} = 3.0$ and set the number of simulations per token to 50 for all the experiments.

For the best setup, we also report the results using T5-large (3 billion parameters), denoted as T5-3B. Using 4 NVIDIA V100 GPUs, $GCN^{\hat{q}=MCTS}$ training took 32 hours for summarization, and 8 hours for QG. This is comparable to the state-of-the-art SelfGAN model. $GCN^{\hat{q}=Nucleus}$ only required 8 hours for training on summarization, and 2 hours for QG.

6.4.2 Results and Discussion

Unconditional Text Generation Figure 6.3 reports results for the unconditional NLG task. First, we observe the crucial importance of the scheduler for the GAN baselines: all of its versions without scheduler (and any normalization as in vanilla discrete GANs) strongly diverge within the first training epoch, obtaining significantly weaker results than MLE (which is the starting point of all curves from the left graph). However, we see that our GCNs are naturally implicitly scheduled, with results comparable to the scheduled version of GANs, thanks to its self-normalized IS. This is an important result since tuning

the rate scheduler from a validation set is tricky and resource-consuming. We also note the significantly better and comparable behavior of $\text{GCN}^{\hat{q}=Nucleus}$ and $\text{GCN}^{\hat{q}=MCTS}$ compared to $\text{GCN}^{\hat{q}=p}$. This validates that the use of smarter sampling helps training, although the space of correct sequences is too large to fully benefit from the MCTS-guided sampling. The right graph from Figure 6.3 plots accuracy vs diversity curves. Here again, we observe the significant impact of scheduling, which is naturally implied in our GCN approach, not only for the sample quality but also on the coverage of the induced distribution. For completeness, the graph also reports curves for previous GAN approaches, including [58, 398], as given by [224] for the same setting. While they are not directly comparable since they do not use the same generative architecture, they all have been shown to fall short compared to their respective MLE counter-part (refer for instance to [49], Figures 3 and 4), which is not the case with our cooperative approach. Note also that SeqGAN with T5 is very similar to $\text{GAN}^{\hat{q}=p}$, using the same kind of incremental discriminator.

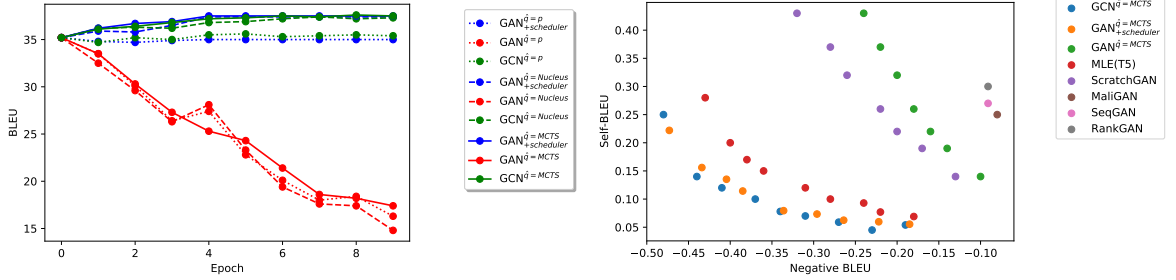


Figure 6.3 – Results on the EMNLP 2017 dataset. Left: Evolution of BLEU results on tests sets w.r.t. training epochs (higher is better) – red: GAN without scheduler, blue: GAN with scheduler, green: GCN. Right: Curves of negative BLEU vs self BLEU (lower is better). Scores for previous studies are taken from [224].

Conditional Text Generation More important are the results for conditional text generation, for which applications are numerous. On both considered tasks, we observe from Figure 6.4 the same trends as for unconditional NLG, with a dramatic divergence of classic GAN approaches. We note a significant improvement of our GCN approaches compared to their GAN scheduled counterparts on both tasks, with a clear advantage for $\text{GCN}^{\hat{q}=MCTS}$ on summarization, where the discriminator guided sampling process obtains very stable results, significantly greater than those of other considered approaches. This result confirms that using MCTS to sample during the learning process is key to producing long texts of better quality.

These trends on the BLEU metrics are confirmed by numerical results from Table 6.1, where $GCN^{\hat{q}=MCTS}$ obtains the best results on both tasks over three metrics, with more than 2 ROUGE-L points gained over the very recent state-of-the-art approach SelfGAN (which also uses MCTS sampling) on QG². Note that these results were obtained without the complex variance reduction techniques that other RL-based GAN approaches require for obtaining results comparable to MLE, which underlines further the interest of our approach. For completeness, we also report results using MCTS for decoding at test time, denoted as $GCN^{\hat{q}=MCTS}_{\text{decod}=MCTS}$, which shows some further improvements, consistently with [304].

Finally, our experiment on scaling $GCN^{\hat{q}=MCTS}$ to a larger model (i.e., T5 3B instead of T5 Small) allows us to further improve the results, indicating the scaling potential for GCN, and establishing a new state-of-the-art for QG and summarization. Please note that, consistently with ColdGAN and SelfGAN, we used a beam-search of size $b = 3$ and no length penalty $\alpha = 0$ (which are set to $b = 4$ and $\alpha = 0.6$ in the original paper T5 paper [271]). Despite these lighter decoding settings, we observe that our $GCN^{\hat{q}=MCTS}_{T5-3B}$ significantly outperforms T5 3B and 11B from the original paper.

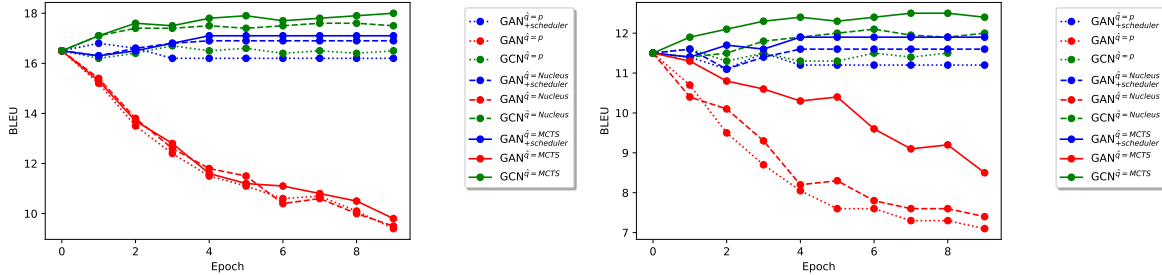


Figure 6.4 – Evolution of performance on the test set w.r.t. training epochs (in term of BLEU, the higher the better), for conditioned NLG tasks. Left: Question Generation, Right: Summarization.

To summarize, we observe that GCN always obtains significantly better results than its GAN counterparts, regardless of the sampling distribution used, in both considered conditional NLG tasks. Moreover, the required tuning of the training scheduler for text GANs is a very difficult task, that implies a very costly grid search process (more than 10 trials was required for each setting of GANs in our experiments), which is not needed

2. Note that, while curves in Figures 6.3 and 6.4 use the MLE pre-train from [305] as the base model, we report here results obtained starting from the best performing MLE model that is used in [304], to obtain results comparable with this paper (results from the former MLE pre-trained model can be found in Appendix B.3).

	QG			Summarization		
	B	R-1	R-L	B	R-1	R-L
MLE	19.7	45.2	41.1	15.9	42.3	40.4
ColdGAN	19.9	45.2	41.4	16.3	42.8	40.7
SelfGAN	20.5	46.6	42.6	17.0	42.8	41.5
$\text{GAN}_{+scheduler}^{\hat{q}=p}$	19.3	45.3	41.2	15.5	40.0	38.8
$\text{GAN}^{\hat{q}=p}$	11.2	26.3	23.9	9.8	23.3	22.5
$\text{GCN}^{\hat{q}=p}$	19.7	46.2	42.0	15.9	40.8	39.5
$\text{GAN}_{+scheduler}^{\hat{q}=\text{Nucleus}}$	20.1	47.3	43.0	16.0	41.8	40.4
$\text{GAN}^{\hat{q}=\text{Nucleus}}$	11.3	26.6	24.1	10.2	23.5	22.7
$\text{GCN}^{\hat{q}=\text{Nucleus}}$	20.9	47.7	44.5	16.6	43.2	41.8
$\text{GAN}_{+scheduler}^{\hat{q}=\text{MCTS}}$	20.4	47.9	43.5	16.4	42.2	40.9
$\text{GAN}^{\hat{q}=\text{MCTS}}$	11.7	27.5	25.0	11.7	24.3	23.4
$\text{GCN}^{\hat{q}=\text{MCTS}}$	21.5	48.3	44.7	17.1	43.4	42.0
$\text{GCN}_{\text{decod}=\text{mcts}}^{\hat{q}=\text{MCTS}}$	21.6	48.7	45.2	17.6	43.7	42.3
$\text{GCN}_{T5-3B}^{\hat{q}=\text{MCTS}}$	21.8	49.8	45.9	19.2	44.2	43.8

Table 6.1 – Final results on QG and Summarization test sets, in terms of BLEU-4 (B), ROUGE-1 (R-1) and ROUGE-L (R-L). Scores in bold are significantly different from the best baseline ($\text{GAN}_{+scheduler}^{\hat{q}=\text{MCTS}}$) according to a 95%-Student-t-test.

in our approach. While the approach requires however to re-sample sequences twice per iteration (cf. line 4 and 6 of Algorithm 2), this additional cost is largely counterbalanced by the ease of deployment and the important accuracy gains. At last, this additional cost can be removed by re-using samples from line 4 at line 7, with an appropriate IS term (no significant accuracy difference in our experiments).

6.5 Conclusion

This work sheds new light on discrete GAN approaches. We give a new perspective to the GAN approach, and introduce a slightly modified algorithm, with strong theoretical guarantees, which can be combined with cooperative sampling strategies to obtain state-of-the-art results on various NLG tasks, using a learned discriminator to drive the generator. This work can be seen as a unification of SelfGAN [304] and ColdGAN [302] to **leverage their benefits (in terms of sample efficiency), while alleviating their drawbacks regarding instability issues, in a theoretically well-sounded framework**. ColdGAN proposed to consider mixtures of behavioral policies, allowing the use of specific - e.g., deterministic - decoding strategies while ensuring that importance sampling holds. SelfGAN proposed to employ an expert-iteration learning scheme, that uses discriminator scores to guide the expert decoding strategy (e.g., MCTS). Our approach leverages

MCTS to drive the sampling distribution with a well-adapted normalization term, which yields **strong implications on the stability of learning**. The reward-augmented MLE obtained using importance sampling weights offers a less-biased gradient estimation than the one of SelfGAN using expert-iteration, whereas the expert decoding creates better samples than the one of ColdGAN. Besides, the introduction of a target distribution, based on discriminator scores, allows to ensure **theoretical convergence guarantees, which were impossible to obtain for both previous approaches**.

For now, we focused on the generator and showed how the information of an external model can be used to enhance the generation (globally or through the addition of constraints). While we did not consider much about the guiding model, we showed that we can generate texts that integrate information from it. Conversely, in the next chapter, we propose to study if information about the behavior of the guiding model can be extracted from such cooperatively generated texts.

THERAPY: GLOBAL EXPLANATION OF TEXTUAL DISCRIMINATIVE MODELS THROUGH COOPERATIVE GENERATION

Contents

7.1	Introduction	94
7.2	Existing Explanation Methods	95
7.2.1	Example-Based Explanations	96
7.2.2	Feature-Attribution Explanations	96
7.3	Therapy	97
7.4	Experiments	99
7.4.1	Experimental Setting	99
7.4.2	Correlation of the Explanations and the Glass-Box Weights	102
7.4.3	Precision/Recall of the Returned Features	104
7.4.4	Insertion/Deletion of Important Features	105
7.5	Conclusion	106

Besides raw accuracy, a critical aspect of models used for misinformation detection is explainability. Indeed, the decisions of such models need to be validated by a human

operator to ensure their validity and prevent the censorship of actual information that is falsely flagged as false. This is especially true for models that are deployed in production and used to moderate content on social media platforms. Any potential bias in the model could be very harmful. For example, a model that would flag as false any content related to a specific minority would be very harmful to this minority. This would also help to ensure that the model uses the raw information in the data to make its decision rather than stylistic clues or generation artifacts. However, the complexity of these models makes it difficult to understand their behavior. Although crucial, the research on the explainability of fake news detection models is limited [233] besides general explanation methods. In this chapter, we introduce a novel method to generate global explanations of any textual discriminative model without requiring input data. This method, called **Therapy**, leverages cooperative generation to generate texts that are representative of the classes learned by the studied discriminative model. The distribution of these texts can then be used to study the classifier and extract the most important words for each class. We show that the method is able to correctly identify the most important words of the model and assign appropriate importance weights in the model prediction. Finally, we show that the terms returned by Therapy are sufficient to modify the prediction of the classifier.

7.1 Introduction

The emergence of machine learning models has led to their broader adoption in domains spanning from mere recommendations to critical areas such as healthcare [47, 98, 168] and law [17, 337]. These already complex models keep becoming larger, emphasizing their black-box denomination. This lack of transparency however slows their adoption in various areas since we witness a significant rise of deployed models suffering from bias. For example, some chatbots biased toward religious [3] and gender [213] minorities have been released and explaining their inner mechanisms is still an ongoing problem.

Among the methods proposed to tackle these problems, the ones that are model-agnostic are preferred since applicable to any machine learning model. Among these, local explanations have obtained strong success by maintaining a good trade-off between accuracy and transparency. These local explanations are generated in the proximity of a target instance by tampering this input to create neighbors and study how the model reacts to these changes. This allows them to highlight which features are important for the model and to provide explanations on the decision for this input (e.g., the most important

words for each class). According to a recent study of the trends in explainability [155], LIME [278], while being the first model-agnostic local explanation method is still the most widely used. However, local explanations has three main flaws when trying to explain a model. First, it obviously requires to have inputs to explain, which might not be possible due to confidentiality and privacy reasons [10] or to protect intellectual property. Second, selecting inputs that are representative of the model or the downstream data distribution is difficult. Finally, it will explain the decision **for this input** and for this input only. This only provides very local information on the model behavior, which represents only a very small piece of the input domain of the model. Therefore, LIME and other local explanation methods have proposed to aggregate the information from multiple samples to provide global explanations. However, these explanations are strongly tied to the input samples and only provide cues about the samples’ neighborhood. These methods thus require samples that cover as much of the space as possible.

To relax this sample dependency and generate global explanations of the model, we propose Therapy, a method that leverages cooperative generation to generate texts following the distribution of a classifier. The distribution of the resulting samples is then used to study which features are important for the model, providing global information on its behavior. We first introduce the usual explanation methods in Section 7.2. We then present Therapy in Section 7.3 and the experiments conducted to compare its performance to standard explanation methods in Section 7.4.

7.2 Existing Explanation Methods

Generating explanations for textual data is challenging since it requires considering both the text semantics and the task domain. Moreover, it is frequent that models are already deployed and further evaluations are required (e.g., fairness, bias detection) but the training data is not accessible. This may be caused by data privacy, security, or simply because the dataset is too large to be analyzed. Thus, to fulfill this objective, researchers have focused on post-hoc explanations [155]. Following the categorization from Bodria et al. [38], we differentiate between example-based and feature-attribution explanations.

7.2.1 Example-Based Explanations

Taking roots from social science [231], the example-based explanations indicate either the minimum change required to modify the prediction –counterfactual– or illustrate class by showing representative instances –prototypes. Counterfactual methods answer the question "what if" and have gained interest since being close to human reasoning, perturbing the document until the model prediction differs [361]. These methods perturb the target document until they find the closest document for which the prediction made by the complex model is different. Conversely, prototype methods select or generate instances that represent the most the target class. Among the example-based methods, some methods propose various control codes to perturb the input text and some others train complex mechanisms to generate realistic sentences based on perturbation in a latent space. Polyjuice [385] and GYC [217] belong to the former and propose control codes varying from changing the sentiment and tense of the sentence to adding or replacing words. On the other hand, xSPELLS [289] and CounterfactualGAN [279] are methods that train respectively a variational autoencoder and a generative adversarial network to convert input text to a latent space and return realistic sentences from this latent space. These methods hence convert the input document into the latent space and slightly perturb it until the closest counterfactual is found.

7.2.2 Feature-Attribution Explanations

Feature-attribution refers to post-hoc explanations methods that associate a weight to input words to indicate the positive or negative impact on the final prediction. Methods such as SHAP [214], LIME [278], and their variants [114, 308, 401, 356, 95, 45] are the most commonly used to generate an explanation [155]. They are local since they perturb an input instance by slightly modifying it and studying the complex model in a given locality. For textual data, LIME randomly masks the words of the input document and trains a linear model on the collection of perturbed documents to predict the decisions of the complex model. The most important coefficients of the linear model associated with the input words are then returned as the explanation. While most explainability surveys [19, 38] are differentiated between local and global explanations, LIME also introduced LIME-SP (for submodular pick), a global method that generates local explanations for a set of n individual instances. These n instances are selected to cover as much of the input domain as possible and avoid redundancy.

7.3 Therapy

We introduce Therapy, a global and model-agnostic explanation method that does not require input data. In place of these input data, Therapy employs an LM guided by the discriminative model D to explain. This cooperation generates texts that are representative of the classes learned by the studied discriminative model. To do so, Therapy extracts the most important words for D by employing it to steer a language model through cooperative generation. Texts generated using cooperative generation follow the distribution $p(x) * p_D(c | x)$. Their distribution can thus be used to study the classifier D : words with high frequencies are likely to be important for the classifier. A logistic regression is then learned on tf-idf representations of generated samples and the weights associated with each term are returned as the explanation. An illustration of the method is proposed in Figure 7.1. Because $p(x)$ is the same for every class, by using tf-idf on the whole corpus (i.e., samples from every class), words that are frequent because of $p(x)$ or in multiple classes will be filtered out. Hence, the logistic regression model learned on the tf-idf score of each feature allows Therapy to study their relative importance and to extract the most important ones for each class. When using MCTS decoding as the cooperative generation method, the method offers the level of explainability of n-grams based on logistic regression models to any classifier. Indeed, since any type of (auto-regressive) language model can be guided during decoding by any classifier using MCTS, the proposed approach is totally model-agnostic. In addition to this plug-and-play property, this approach exhibited state-of-the-art results in the task of constraint generation, that is, generating texts that maximize $p_D(c | x)$ while maintaining a high quality of writing. We thus experiment with MCTS decoding for Therapy, but the proposed method is compatible with any cooperative generation approach.

In essence, the method is similar to using LIME jointly with a masked language model to generate neighbors when the number of replaced tokens grows a lot but with two benefits. First, the method does not rely on input examples but creates samples out of nothing using the language model. This is useful for cases where the data cannot be shared because it contains confidential information [10]. Moreover, rather than exploring the neighborhood of these examples (and so conditioning the explanations on these examples' context), the domain of the exploration is defined by the domain of the language model, which is significantly broader. Besides, either a general language model can be used to study the model behavior on generic data or a language model specific to the downstream domain

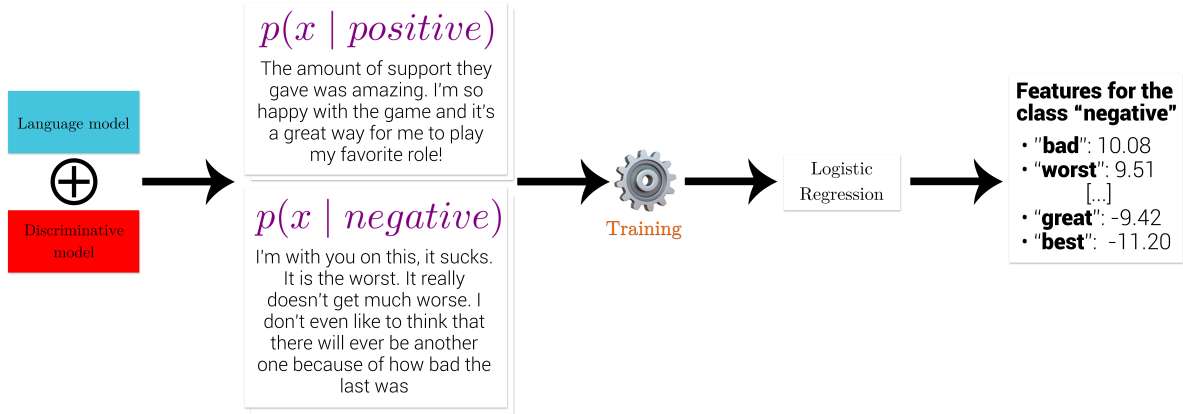


Figure 7.1 – Illustration of the Therapy method. Texts from different classes are cooperatively generated using the guidance of the studied model. A logistic regression is then trained to predict the label of the generated texts. The weights of the model associated with each word are then returned as importance weights.

to make sure it works well on this specific type of data.

Second, the method does not generate **before** classifying the text but employs the classifier **during** the generation. Hence, instead of "randomly" generating texts and hoping for important features to appear, we explicitly query the model for stereotypic features by maximizing $p_D(c \mid x)$. This makes the method more efficient and reduces the probability of generating rare features that are not important for the model while reducing the odds of generating "in the middle" texts that have features from multiple classes and are misleading. Besides, our method directly relies on the distribution learned by the studied model to guide the generation, unlike methods like Polyjuice and GYC, which, in addition to requiring input data, count on a distribution learned by the language model to bias the generation towards the desired property (using control codes).

Finally, Therapy is distinctive from methods analyzing the frequency of input terms in the training data such as sensitivity analysis since it does not require access to (training) data and directly exploits the distribution effectively learned by the model, whereas nothing guarantees that a model is actually using the terms extracted from training data to make a prediction. Furthermore, our method differs from existing example-based and feature attribution methods since to the best of our knowledge, there exists no global and model-agnostic explanation methods that do not require any input data.

We call this approach Therapy because its functioning is similar to that of a therapist.

This therapist (the LM) queries its patient (the classifier) to understand its behavior and eventually discover pathologic behaviors (some biases).

7.4 Experiments

In this section, we first give technical details on the experiments conducted to evaluate Therapy (Section 7.4.1). We then evaluate Therapy through three experiments. The first one (Section 7.4.2) measures the Spearman correlation of the explanations and the weights of a glass-box and studies the influence of the number of generated texts on the quality of the explanation returned by the linear model. We then compare the capacity of the method to correctly identify the most important words of the glass-box to the one of LIME and SHAP using precision/recall curves in Section 7.4.3. Finally, we test whether the terms returned by the different approaches are sufficient to modify the prediction of the classifier in Section 7.4.4.

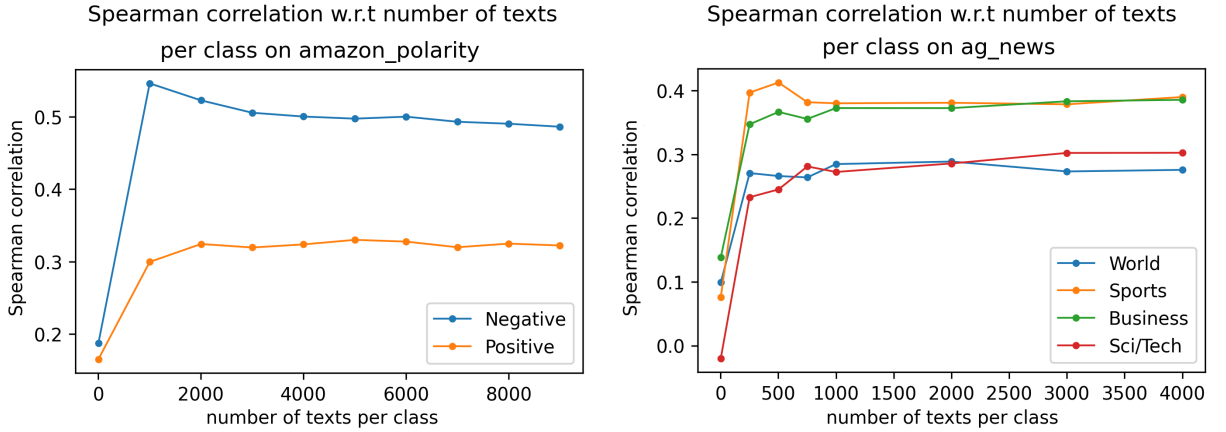


Figure 7.2 – Spearman correlation w.r.t number of generated text per class for amazon_polarity and ag_news.

7.4.1 Experimental Setting

Glass-box explanation Since there are no ground truth explanations available to be used as a goal for evaluated methods, we use a glass-box model, that is, a model explainable by design but used as a black-box (i.e., without being able to use its inner workings to generate explanations). Following prior work [125], we train a logistic regression using `sklearn` [256] and use its weights as tokens importance scores.

Therapy implementation To evaluate the proposed method, we use the [implementation of PPL-MCTS](#) of Chapter 5 and simply plug the glass-box by defining the function that takes a sequence and returns its score. The choice of the LM to guide defines the domain on which we want to explain the behavior of the model. Thus, it is best to choose a language model that is as close as the domain on which the discriminator will be used. However, to show that the proposed approach works well, even on a general domain, we use [OPT-125m](#) [409]. A logistic regression is then learned on generated texts and its scores are used as token importance.

Datasets Experiments are conducted on two previously introduced classification datasets from [410], [amazon_polarity](#) and [ag_news](#). The reviews of [amazon_polarity](#) are rather small and have highly caricatural lexical fields, while articles in [ag_news](#) are longer and more diverse but include distinctive indicators because they are extracted from online news articles. Samples generated by Therapy along with the first 20 top words returned by the method for both classes of [amazon_polarity](#) is given in Figure 7.3. Please note that some "words" correspond to sub-words, due to the breakdown into unigrams (ve, ll, ...). The proposed approach allows to study the impact of n-grams, but this is not possible with LIME and SHAP (using available code), so we restricted the study to unigrams. Samples and top words of [ag_news](#) are given in Appendix C.1.

Compared methods In our experiments, we compare the results of Therapy to the two most widely used post-hoc methods: [LIME](#) [278] and [SHAP](#) [214]. We employed publicly available implementations of these traditional methods instead of their extensions mentioned in Section 7.2. This decision was made because, to the best of our knowledge, these extensions either do not prioritize the generation of global explanations or do not enhance the textual versions of these methods. The main difference between LIME and SHAP is that the former generates samples by modifying input data and then learns a linear regression model whereas the latter benefits from game theory to compute the weight of each term. We use the global version of these methods on 500 texts of the datasets test set. For SHAP, we keep the 10 000 most important words for each dataset whereas, for LIME, we computed 500 local explanations with the 35 most important words and merged every term-weights pair into dictionaries of length 4592 for [amazon_polarity](#) and 5770 for [ag_news](#). Finally, to highlight the benefits of cooperative generation in Therapy, we also report the results obtained by a simple baseline. Rather than using cooperatively generated

Class "positive"**Samples:**

- "The amount of support they gave was amazing. I'm so happy with the game and it's a great way for me to play my favorite role! Also, I love that they're giving us a new character every week, which is awesome :)I think they've been pretty good lately too. It's nice when we see someone who really loves their team and has fun doing things"
- "The whole point of this post is to show people what a great job you've put in. You are doing an excellent job, I love this sub. Keep going. :)I'm so happy for all those people that have helped! It's amazing to be here and it was fun being the one that got the most attention on the other day :). Thank you so much <3You're very welcome :) And yes... I am glad you enjoyed this subreddit - it is a great place"

Top words: *great, love, good, ve, years, people, lot, friends, fun, life, world, works, easy, things, happy, heard, including, awesome, nice, family*

Class "negative"**Samples:**

- "I guess you could say it has to be the most boring thing ever.It is a very boring experienceI don't think they have any idea how to make money off this, so they will probably just do what you say they will if it doesn't work. They may try to sell you on the fact that it isn't working and then try and get you to buy back into it when they are done making money. If I was them, would they not want to pay me?"
- "It might be a bug, but it doesn't work on my phoneYeah that sucks! I was hoping the same thing happened.I don't think you can get a refund without having to buy a new one...You have no idea what they would be like without the warranty. They wouldn't know how much money is left if they didn't do it properly or at all and they would probably just give it back as soon as they found out it was defective (or something)"

Top words: *don, money, bad, doesn, didn, idea, work, device, isn, thing, guess, wrong, back, buy, fact, time, phone, point, problem, thought*

Figure 7.3 – Samples generated by Therapy and top words returned by the method for both classes of the amazon_polarity dataset.

Dataset	AMAZON__POLARITY		AG__NEWS			
Class	Positive	Negative	World	Sports	Business	Sci/Tech
Baseline	0.49 (6.24e-08)	0.31 (9.25e-05)	0.25 (1.67e-06)	0.32 (6.58e-09)	0.35 (1.88e-11)	0.12 (2.33e-02)
Therapy - most played	0.52 (5.79e-09)	0.32 (7.83e-05)	0.22 (1.57e-05)	0.27 (7.66e-07)	0.32 (2.04e-09)	0.22 (1.93e-05)
Therapy - highest score	0.49 (3.3e-08)	0.31 (1.0e-04)	0.27 (1.6e-07)	0.37 (4.0e-12)	0.38 (5.6e-13)	0.3 (8.9e-09)

Table 7.1 – Spearman correlation (p-value) between the top words of a logistic regression glass-box and explanation methods learning a logistic regression over generated texts. Baseline uses unconstrained samples while Therapy generates samples using the MCTS, either selecting the most played or highest scored node. Results are shown per class and dataset.

Dataset	AMAZON__POLARITY		AG__NEWS			
Class	Positive	Negative	World	Sports	Business	Sci/Tech
Baseline	0.49 (6.24e-08)	0.31 (9.25e-05)	0.25 (1.67e-06)	0.32 (6.58e-09)	0.35 (1.88e-11)	0.12 (2.33e-02)
LIME	0.64 (5.0e-7)	0.44 (1.5e-3)	0.09 (0.53)	0.16 (0.27)	0.20 (0.16)	0.19 (0.19)
LIME-other	0.21 (0.14)	0.18 (0.21)	-0.03 (0.85)	0.23 (0.12)	0.09 (0.52)	0.29 (0.04)
SHAP	0.71 (7.6e-9)	0.76 (1.6e-10)	0.47 (6.2e-4)	0.62 (1.7e-06)	0.53 (8.0e-5)	0.61 (2.4e-6)
SHAP-other	0.02 (0.87)	0.26 (0.06)	-0.05 (0.71)	0.04 (0.77)	0.15 (0.31)	0.12 (0.41)
Therapy	0.49 (3.3e-08)	0.31 (1.0e-04)	0.27 (1.6e-07)	0.37 (4.0e-12)	0.38 (5.6e-13)	0.3 (8.9e-09)

Table 7.2 – Spearman correlation (p-value) between the top words of a logistic regression glass-box and the four explanation methods. ‘other’ indicates that the explanations are generated using the other dataset. Results are shown per class and dataset.

texts to train the logistic regression, the baseline generates texts without constraining the language model and uses the glass-box **after** the generation is done to get the target labels.

7.4.2 Correlation of the Explanations and the Glass-Box Weights

A good explanation of the glass-box is a list of features that contains both its important features (i.e., has good coverage) and links them to a similar relative weight. Hence, we compute the Spearman correlation between the top words of the glass-box (having a weight > 1) and their scores attributed by the explainer. We selected Spearman correlation over Pearson because the score returned by LIME and SHAP can be very different from logistic regression weights and so rank correlation results in a fairer comparison.

Influence of the number of generated texts One critical parameter of the proposed method is the number of texts to generate since more tokens allow a larger coverage but

require more computation. We report the Spearman correlation against the number of generated texts per class in Figure 7.2. We observe that the correlation quickly rises until plateauing, meaning that only a small amount of text offers a great overview of the model behavior and that the method does not require a lot of computing to perform. We thus fix the number of generated texts for Therapy to 3000 for each class for the rest of our experiments.

Importance of the classifier guidance Cooperative generation allows Therapy to guide the language model during the decoding process and to move away from its distribution toward that of the model studied. To study the importance of this guidance, we report, in addition to the baseline, the results obtained when selecting the most played token during MCTS generation. As mentioned in Section 5.2, the token added to the current context can be selected as the most played node or the one obtaining the highest score. Selecting the highest-scored node generate texts that are the most stereotypical of the studied model, while the most played node is closer to the language model a priori. Results reported in Table 7.1 show that both the baseline and using the most played node exhibit competitive results on `amazon_polarity` but struggle more on `ag_news`. This can be explained by the fact that the language model tends to not generate positive and negative terms at the same time, so the classes are clearly defined even in unconstrained samples. On `ag_news`, however, there is more overlap between classes, and so using cooperative generation helps to generate texts that are more distinctive of a given class. These results both highlight the contribution of the cooperative generation and motivate the token selection method.

Comparison with other methods The Spearman correlations of all the evaluated approaches can be found in Table 7.2. Results yielded by Therapy are better than those of LIME on `ag_news` but worse on `amazon_polarity` whereas SHAP yields better results than both methods on both datasets. Counterintuitively, these are positive results for Therapy because other methods have access to the test set of the studied dataset, ensuring that the target features are found in the input examples. To test the performance when this assumption no longer holds, we resort to two variants of LIME and SHAP, denoted by *-other*. The key distinction between these methods lies in the dataset employed as input data. We use `amazon_polarity` texts as input to find features in `ag_news` and vice-versa. The findings from these experiments reveal that existing methods fail to find important features, leading to a significant drop in correlations, substantially lower than those of

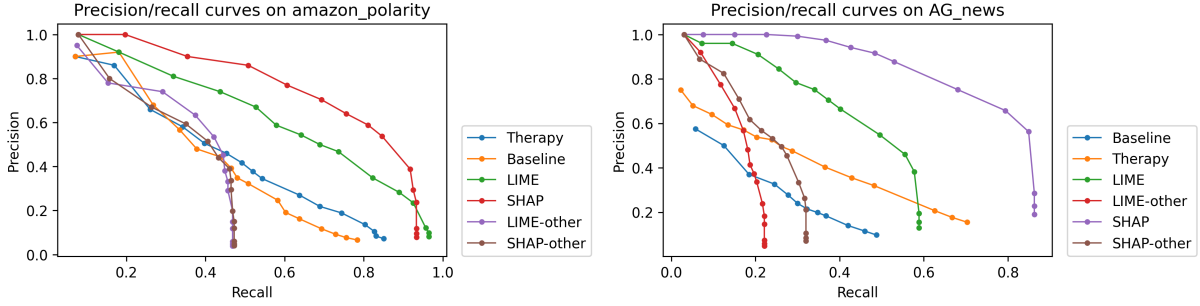


Figure 7.4 – Precision/recall curves of the glass-box top words for the different explanation methods.

Therapy.

7.4.3 Precision/Recall of the Returned Features

Besides assigning correct scores to important features of the model, we also want to make sure that Therapy gives an informative output in practice. That is, making sure that most features returned by the explainer (i.e., its highest-scored features) are indeed important features of the original model and that most of its important features are found. Thus, we report precision/recall curves averaged over every class in Figure 7.4. Precision is obtained by computing, for different numbers of words returned, the proportion that is in the most important features of the original model. Conversely, recall is the proportion of the original model’s top words retrieved. The number of words returned ranges from 10 to 1500.

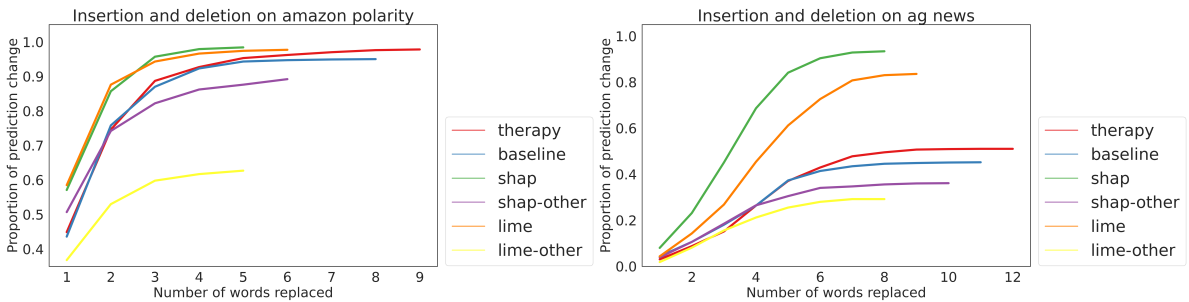


Figure 7.5 – Proportion of texts whose glass-box prediction changes w.r.t the number of important words from the original class replaced by important words from other classes.

Therapy yields worse results than LIME (although achieving better recall on ag_news)

and SHAP on both datasets. Again, when the input data does not necessarily contain the important features for the model (-other), the results collapse and Therapy outperforms both approaches. The limitation of these methods is visible by the plateau in their recall scores: they indeed find the important features **present in the data**, but are limited to these only, setting the upper limit of features that can be found. In practice, biases contained in the model can be subtle enough not to be present in the available data, in which case LIME and SHAP will not be able to detect it. Therapy, on the other hand, obtains good results while using the same generic LM for both datasets, without using any a priori. The method thus provides a very good overview of the model’s behavior when no data, or more broadly, when no data representative of the important features of the model is available. In the latter case, Therapy offers a broader search than the one based on existing texts, offering higher recalls.

Again, the baseline is competitive against Therapy on `amazon_polarity` but is significantly worse on `ag_news`. This illustrates that the cooperative generation allows Therapy to better highlight distinct classes when they are more mixed in the language model.

7.4.4 Insertion/Deletion of Important Features

A strategy to validate the correctness of the explanation is to remove the features that the explanation method found important and see how the prediction of the model evolves. The intuition behind deletion is that removing the “cause” will force the model to change its decision [262]. Similarly, adding a word returned by the explanation as important for another class should lower the confidence of the model. Thus, we compute an insertion/deletion metric that measures the proportion of texts whose glass-box decision change when a word listed as important for the original class is removed and replaced by an important word from another class. Figure 7.5 shows the results on both datasets for Therapy, the baseline method, LIME, SHAP, and their version using the other dataset as input (-other) on 1000 texts. Replacements are done by iterating over the list of the top 250 words returned by each method for the original class until the decision of the model changes. Replacement can only occur if the word is present within the text and multiple replacements of the same word in a given text are counted as multiple replacements. This explains why each method has a different maximum number of words replaced. Methods that leverage generative models seem to achieve more replacements. We hypothesize that this is because they are designed to globally explain the model on the input domain, unlike local methods that can return words that are specific to a given input and not generalize

well.

We observe that Therapy achieves very similar results to those of LIME and SHAP on amazon_polarity but significantly worse than both on ag_news. However, when compared to the -other versions, Therapy achieves very convincing results, showing once again that these methods require very specific data while Therapy is able to find important words for each class without accessing any data nor using any a priori on the model. In this experiment as well, Therapy outperforms the baseline on both datasets, although the difference is more noticeable on ag_news.

7.5 Conclusion

Usual explainability methods rely heavily on input data, which is not necessarily available and might not contain model biases or important features. We propose Therapy, a method that leverages cooperative textual generation to create synthetic data that follow the studied model distribution. Thus, **the search of important features is driven by a pre-trained language model rather than input samples**. The pre-trained language model allows a **broader exploration** than being restricted to input data neighborhood, relaxing most of the constraints and a priori induced by examples-driven methods. In the extreme case where extremely representative data (such as the test set of a given dataset) of important features of the model is available, Therapy lags a bit behind state-of-the-art SHAP while being competitive. However, **when considering more realistic cases where we do not explicitly give the important features to the explainer or do not have any available data, its performances are very good whereas the other methods are collapsing when even applicable**. Comparisons with a generate-then-classify baseline highlight the benefits of the cooperative generation when the language model does not generate texts that are representative of a single specific class by itself.

Therefore, Therapy is a useful tool to **explore the model behavior on a large domain when collecting data that exactly match the downstream distribution is not feasible**. For example, it can be used to detect stylistic biases leveraged by existing misinformation detection models and highlight words that guide the decision including those that should not be useful for the task. Besides, we opposed the proposed approach to LIME and SHAP to highlight the interest of generating representative texts using cooperative generation when input data is lacking. However, an interesting avenue of research would be to use these established explainability methods on cooperatively generated texts,

replacing the proposed logistic regression on the tf-idf representations. This potential combination might allow to leverage their performance while alleviating the input data dependency.

Finally, while our experiments focused on textual classifiers, the proposed approach is compatible with any discriminative models that have text in their input and output a score, including cross-modal retrievers as discussed in Chapter 10. Although very similar to the one of PPL-MCTS, we make the code of Therapy [publicly available](#) to allow the community to quickly explore their models and study the use of cooperative generation as an explainability approach.

After exploring different use cases of cooperative generation, we explore in the next chapter the computational overhead of cooperative approaches using different kinds of Transformer-based guiding models that yield different complexity/quality trade-offs.

WHICH DISCRIMINATOR FOR COOPERATIVE TEXT GENERATION?

Contents

8.1	Introduction	109
8.2	Choosing the Right Teammate	109
8.3	Empirical Study	111
8.3.1	Discrimination Strength	112
8.3.2	Generation Quality	113
8.3.3	Computational Gain	114
8.4	Conclusion	116

We showed in previous chapters that cooperative generation can be used to generate texts satisfying a constraint defined by an external classifier, to train a generative model using cooperatively generated texts guided by a discriminator and even to explain the behavior of the guiding model. One natural question that arises is the computational overhead brought by this kind of approach, especially when the external model is used **during the decoding at inference time**. In this chapter, we explore the impact of the choice of the guiding model on the guidance quality and the computational complexity of cooperative generation.

8.1 Introduction

Currently, top performing discriminators are Transformers using bidirectional attention [88, 209, 135], but, as introduced in Part 1, this does not fit the iterative nature of the generation process. Indeed, since each token attends not only to previous but also to future tokens in the sequence, it requires to recompute every hidden state of the whole sequence for any additional token. This prevents the use of cached hidden states and results in a quadratic cost w.r.t. the sequence length of attention layers at each timestep. On the other hand, unidirectional Transformers employ left-to-right masks to only depend on past tokens for text encoding/decoding [269]. Thus, a new token does not change the representation of previous tokens, so hidden states can be reused for subsequent steps, hence involving linear computing complexity (attention scores of this new token). However, these two types of discriminators only score one sequence at a time, given as input of the model. This limits the number of possible tokens to be considered at each decoding step, to avoid a computationally prohibitive cost. Solving this issue, generative discriminators [180] give scores for all tokens from the vocabulary at once, hence dramatically reducing the cost of width exploration. In this chapter, we explore the pros and cons of these three types of discriminators (bidirectional, unidirectional, generative) when used in cooperative language decoding. We conduct our experiments on the Monte Carlo Tree Search cooperative decoding introduced in previous chapters and provide an [implementation of the MCTS that allows to generate texts in batch for each type of discriminator](#) based on the HuggingFace’s Transformers library [382].

8.2 Choosing the Right Teammate

Attention layers as defined in [353] are bidirectional: every token can attend to tokens at every position. Transformer decoders used for text generation, on the other hand, leverage unidirectional attention masks [269]. Models using bidirectional attention are commonly used for discriminative tasks because they are expected to be strictly more powerful than their unidirectional counterpart [88]. This higher capacity comes at the cost of the auto-regressivity: since the tokens attend to the next tokens, any additional token changes the representation of the whole sequence. This prevents the use of cached hidden states, inducing a quadratic cost w.r.t. the sequence length at each timestep. In the unidirectional setting, any extra token added at the end of a sequence does not change the

already calculated hidden states, since previous tokens do not attend to this new token. Thus, starting from an already classified sequence $x_{1:t-1}$, classifying $x_{1:t}$ only requires computing t attention scores, rather than the whole set of t^2 scores per self-attention layer, as would be required in the bidirectional setting. In common discriminative tasks, this does not matter since only entire sequences are discriminated. Hence, none of the hidden states needs to be reused for another next sample. However, for a use in auto-regressive cooperative decoding, where input sequences are the continuation of already discriminated ones, unidirectional attention allows to reuse the contextual encodings of previous tokens, hence greatly speeding up the process.

However, as mentioned earlier, evaluating every possible continuation of a given sequence is intractable, even with unidirectional discriminators, since for a vocabulary of size $|\mathcal{V}|$ it requires $|\mathcal{V}|$ forward passes at each decoding step. $|\mathcal{V}|$ being in the order of ten thousand, discriminating every possible continuation of decoding sequences is too costly. Thus, cooperative approaches have to circumvent this issue by limiting the number of sequences actually evaluated by the discriminator, for example [303] pre-filters potential continuations on the nucleus of the LM distribution [144]. This choice necessarily biases the resulting generated distribution.

Generative discriminators [180], on the other hand, exploit class-conditional language models [172] to discriminate every token at once. CC-LMs condition distributions of sequence x on a desired class of interest c : $p(x | c) = \prod_t p(x_t | x_{1:t-1}, c)$. Assuming a uniform prior distribution of classes $c \in \mathcal{C}$, Bayes' rule enables to use this for discrimination: $p_D(c | x_{1:T}) \propto p(x_{1:T} | c)$ as illustrated in Figure 4.1. Thus, it only requires $|\mathcal{C}|$ forward passes to get the discrimination scores of all possible sequence continuations. $|\mathcal{C}|$ being usually much lower than $|\mathcal{V}|$, this makes the consideration of every token tractable for sequential discriminative decoding. To improve the discriminatory capacity of such models, training of CC-LMs used in GeDi leverages a discriminative loss \mathcal{L}_d in addition to the traditional language modeling loss \mathcal{L}_g . This discriminative loss corresponds to a cross-entropy loss using the model as a discriminator and a hyper-parameter λ is used to define the balance between the two objectives: $\mathcal{L}_{total} = \lambda\mathcal{L}_g + (1 - \lambda)\mathcal{L}_d$.

These three types of discriminators offer different capacity/complexity trade-offs, which are studied in this chapter for cooperative decoding with MCTS. More precisely, three questions are explored:

- How these models differ in pure discrimination accuracy?

- To what extent are these differences noticeable in generated texts?
- How do these methods compare in terms of computation complexity for cooperative decoding?

8.3 Empirical Study

According to previous studies, unidirectional models should yield worse accuracy than bidirectional ones [88, 269] and better than discriminative generators [393, 241]. To thoroughly assess the pros and cons of these models using state-of-the-art Transformer architectures, the only difference must be the studied property (uni- vs bi- directionality, and discriminative vs generative). Thus, we propose to use the same backbone for all settings to prevent any external confounding factors, with a single fully connected output layer on top of the contextual embedding of the last token to produce discrimination scores. Starting from BERT [88] as the bidirectional discriminator, a triangular self-attention mask is applied for adapting it from the bidirectional to the unidirectional setting in our experiments, following [90]. Then, the generative discriminator is the same as the left-to-right one, the only difference being the size of the output layer that changes from $(hidden_size, num_classes)$ to $(hidden_size, vocab_size)$.

As in the previous chapter, experiments are made on two datasets from [410]: `amazon_polarity` and `AG_news`. These datasets allow to study the results of cooperative generation on two rather different constraints and domains: applying polarity to online reviews and writing news about a specific topic. Also, `AG_news` allows to study the generalization to non-binary classification and texts with more diverse content. Each model is trained for 20 epochs using AdamW [210] with HuggingFace’s trainer default parameters ($\beta_1 = 0.9, \beta_2 = 0.999, eps = 1e-8$) and a linear scheduler with no warmup. Batch size is set to the maximum that can fit in the memory of a Quadro RTX 6000 during GeDi training (4 for `AG_news` and 8 for `amazon_polarity`). Gradient accumulation is set to emulate a batch size of 128. For training GeDi, we set $\lambda = 0.6$ according to the authors (and did not observe a significant difference when setting $\lambda = 0$ to strengthen the classification capacity).

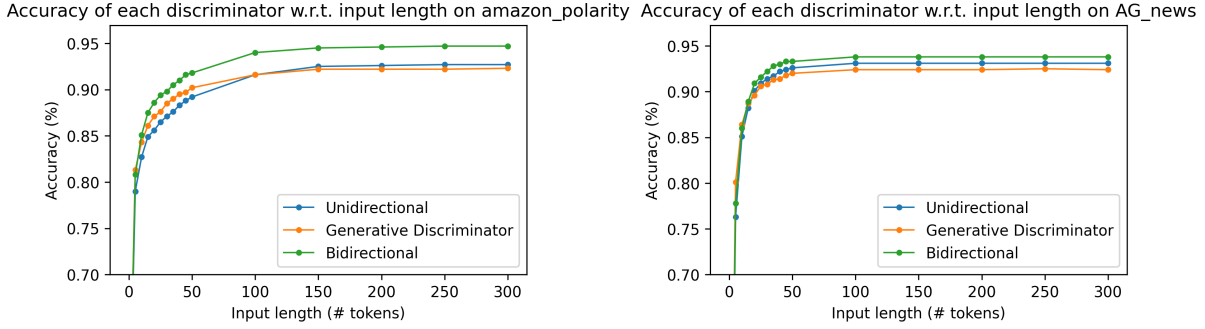


Figure 8.1 – Accuracy (%) of the different type of discriminators w.r.t. the input length (# tokens)

8.3.1 Discrimination Strength

For discriminators, accuracy is of the utmost importance: it defines how well it solves the intended task. In the context of cooperative generation, having a good accuracy on complete sequences is not sufficient: an informative output with incomplete sequences is needed so that the discriminator can be used throughout the generation process. Thus, plotting the accuracy w.r.t. the number of input tokens gives information about the capacity of the model to guide the generation at different timesteps, the main property expected for discriminators in cooperative generation. Note that, the discriminator is trained on sequences of variable lengths to avoid a mismatch between training and test tasks.

Results reported in Figure 8.1 show that every discriminator exhibits the same behavior: starting from random predictions, accuracy quickly increases with the input length until reaching a plateau. The expected ordering is observed: bidirectional models perform better than unidirectional models, which themselves perform better than generative ones. However, it should be noted that the gap is rather small and only appears when approaching the plateau. Favoring bidirectional models in accuracy-critical tasks is justified, but it is not necessarily clear that these small differences will reflect in the quality of cooperatively generated texts.

Please note that this corresponds to the accuracy on *in domain* data, and that complementary - non-reported - experiments, on random sequences to be discriminated, showed however that GeDi is more robust to *out of domain* sequences: its discrimination scores are greatly closer to maximal uncertainty (i.e., $p_D(c | x) = 0.5$) than those of discriminative models which tend to greatly favor one class over the other ones in such cases. However,

this may not impact the results of cooperative generation, since such random samples are not likely to be observed during MCTS decoding, because of the language model prior guiding search toward in-distribution sequences.

8.3.2 Generation Quality

To assess whether the – relatively small – differences in classification accuracies impact the results on cooperative generation with MCTS, we follow the PPL-MCTS setup from Chapter 5 by constraining the generation process towards a desired class c using $p_D(c | x)$ given by the considered discriminator. The same automatic metrics are used to study the quality of the guiding signal brought by the discriminator: 1) Accuracy corresponds to the average rate of generated sequences for any class c to be correctly classified as c by an oracle discriminator trained on disjoint data, 2) Self-BLEU [426] focuses on diversity across samples, by measuring BLEU scores between generated sequences, and 3) Oracle perplexity stands for the perplexity of an oracle LM trained on disjoint data, allowing to control the writing quality of generated texts. We used a bidirectional BERT model as the oracle discriminator to get the most accurate evaluation possible. Language models are also BERT models with an LM head in order to use the same tokenizer. Average results over 500 sampled test texts using each type of discriminator on the two datasets are reported in Table 8.1. We also report results obtained using the vanilla LM likelihood $p(x)$ as the back-propagated score in MCTS evaluation, to provide baseline results achievable without discriminators. Results are obtained using best-performing hyper-parameters ($c_{puct} = 3$, temperature $\tau = 1$) and 50 iterations of MCTS per token unless specified otherwise. We report statistical significance between each type of discriminator using a t-test with p-value=0.01.

The difference in generation accuracy when using bidirectional and unidirectional discriminators shows that the difference in raw accuracy reflects in resulting samples when used for cooperative generation. The higher difference on amazon_polarity also results in a higher difference in generation accuracy. However, this difference is relatively limited and the generation does not seem to deviate too much when using unidirectional discriminators. Results using generative discriminators are different, with a significantly greater drop of accuracy than between uni- and bi-directional models on AG_news, although the gap in raw accuracy is similar. More surprising is the result on amazon_polarity where, despite similar raw accuracies, we observe a 10-point drop in generation accuracy. We hypothesize

that this is because the signal is not as informative: while raw accuracies are pretty similar, the average score attributed to the ground truth class in evaluation is significantly lower in the case of GeDi. This means that its signal promotes less the good solutions than standard discriminators when guiding the generation. The type of discriminator has no significant impact on the other metrics. Please note that the general difference in Self-BLEU and oracle perplexity between the two datasets is due to the difference in their content: AG_news is more diverse, which results in lower Self-BLEU and higher perplexity. Finally, we notice that doubling the number of MCTS iterations allows to increase the accuracy results of the unidirectional model, bridging the gap between both models for a still lower computational cost (see next section).

Value	amazon_polarity			AG_news		
	Accuracy \uparrow	5 - Self-BLEU \downarrow	Oracle perplexity \downarrow	Accuracy \uparrow	5 - Self-BLEU \downarrow	Oracle perplexity \downarrow
$p(x)$	70.8	0.652	10.49	86.6	0.306	29.08
Bidirectional	96.0*	0.531*	12.25	94.8*	0.319	29.13
Unidirectional	93.0*	0.528*	11.98	93.4	0.313	29.99
Unidirectional (100 its)	93.6*	0.522*	10.73	94.6*	0.323	30.92
Generative discriminator	84.4	0.576	11.92	91.8	0.321	29.43

Table 8.1 – Performance of MCTS w.r.t. the metric to optimize on amazon_polarity (left) and AG_news (right) datasets. * indicates statistically significant improvement against Generative Discriminator. Note that no model demonstrated significant improvement over the unidirectional discriminator.

8.3.3 Computational Gain

Beyond generation accuracy, we are interested in the computational complexity of the various models to be used in cooperative generation. Figure 8.2 reports MCTS execution times w.r.t. each generation step t (i.e., the time required to decode token at step t of any sequence), using a bidirectional model compared to a unidirectional one. Unsurprisingly, since the complexity is quadratic in the bidirectional case and only linear in the unidirectional one, the difference in generation time is significant and increases linearly w.r.t. the sequence length. Note also that this difference increases with the number of MCTS iterations. At last, we note that the number of MTCS iterations with unidirectional discriminator can be much more than doubled compared to the case of bidirectional one, while keeping the computational cost significantly lower, even for small text sequences.

In the case of the generative discriminator, a great potential computational gain may arise from the fact that discrimination scores can be computed for every child of an

MCTS execution time (s) w.r.t. generation step on amazon_polarity

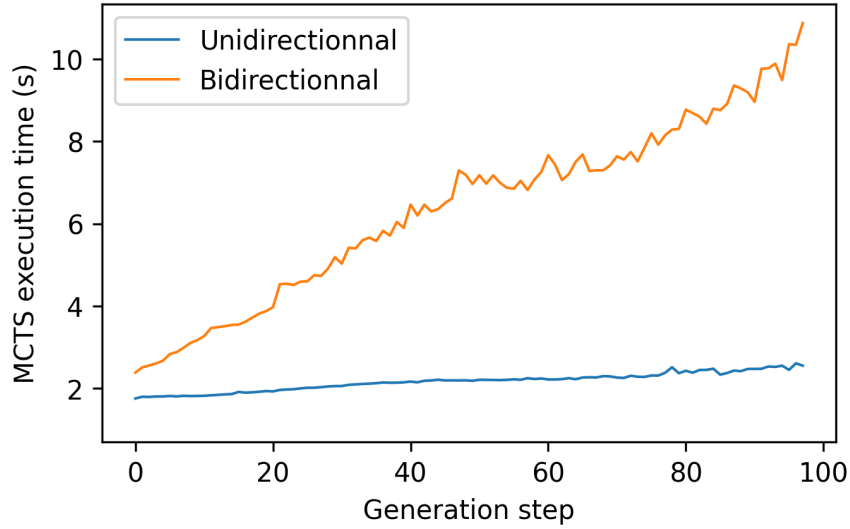


Figure 8.2 – Execution time of MCTS iterations (s) w.r.t. generation step (averaged over 10 batches of 30 sequences)

expanded node at once. More specifically, while computing scores for each of the $|\mathcal{V}|$ children nodes would cost $|\mathcal{V}|$ forward passes in the case of discriminative classifiers, it only requires $|\mathcal{C}|$ forward passes for generative classifiers (i.e., one pass per class for getting all scores, rather than one pass per child node). Since usually $|\mathcal{C}| \ll |\mathcal{V}|$, the use of a generative discriminator could be way advantageous and allow to increase the number of MCTS iterations to expect to, at least, fill the gap with accuracy results of discriminative approaches.

However, this potential gain heavily depends on the exploration of the tree and therefore the parameter c_{puct} . If less than $|\mathcal{C}|$ children are considered at each level of the tree, then the generative approach is at least as costly as the discriminative one and can even be more costly. Indeed, we empirically observed that for the usual value $c_{puct} = 3$, the generative discriminator needs in average 1 685 more forward passes on amazon_polarity (where $|\mathcal{C}|$ is only 2), meaning there is more depth than width explorations. Increasing c_{puct} decreases this difference but also the resulting generation accuracy. At $c_{puct} = 15$, the accuracy already drops by 10 points and the difference is still to the disadvantage of GeDi for more than 600 forward pass. These results show that generative discriminators are only beneficial if exploration is wider than deeper, which is not the case for MCTS operating points. This is consistent with GeDi results [180], which observed an important gain in

a beam search decoding approach where the width is crucial. These new results suggest seeking ways for better leveraging this GeDi potential with more efficient exploration in width of the MCTS or to use methods that do it by construction such as beam search.

8.4 Conclusion

Cooperative generation has proven to be an effective way to augment traditional text generation with external information from a discriminator. While Transformers with bidirectional attention are usually preferred for discriminative tasks, **they are not autoregressive and are therefore much more expensive when used to guide generation**. Although a little less precise, **unidirectional Transformers allow to achieve very similar results for a much more reasonable and consistent cost**. As a consequence, our study shows that unidirectional discriminators should be preferred for cooperative generation, for which slight accuracy drops can be balanced by reinvesting part of the computational gain. Given the size of usual vocabularies, **generative discriminators seem very interesting at first glance to allow a wider search**. However, while achieving similar results in terms of classification accuracy, scoring the whole vocabulary comes at the price of a **less informative signal**. Moreover, although counter-intuitive, **this width is not necessarily useful** as shown by the search performed by the Monte Carlo Tree Search, which usually explores more in depth than in width. The distribution of the **language model is indeed expected to already assign most of the probability density to the few tokens that can fit in the sequence**. Thus, such models will prove useful when used with **methods that make particular use of this width information**. We leave such explorations for future work. To allow these further experiments as well as reproduction, the code used for our experiments is made [available for the community](#).

After thoroughly exploring different aspects of cooperative generation on NLP tasks, we introduce how this approach can be used to improve the results of multimodal misinformation detection by enhancing the cross-modal alignment and retrieval.

PART III

CROSS-MODAL GENERATION

CROSS-MODAL GENERATION AS AN ALIGNMENT TASK

In the previous parts, we extensively studied textual generative models while our interest lies in being able to discriminate multimodal input data to detect misinformation. Considering our goal of creating a whole class made of synthetic examples, samples generated using cooperative generation manage to fool the discriminator at a given timestep. Unfortunately, it manages to adapt very quickly and achieves a high detection rate again, even for generators trained using GCN. Even though generating more realistic sequences is useful for different data augmentation techniques, as the one presented in Chapter 2, generated samples still have too many generation artifacts and biases to be used as a proper class. Yet, besides data augmentation, advances in generative models can still improve the performance of multimodal misinformation detection. Notably, we now focus on cross-modal retrieval, a task that allows seeking external pieces of evidence that can serve to verify the input information and is really effective against image repurposing [2]. The information retrieved can be used in knowledge-based approaches, notably as input for retrieval-augmented language models [130, 195, 42, 154] and be forwarded to a human fact-checker to add a layer of explainability and help assess the veracity of the information to make the final decision faster.

9.1 Multimodal Transformers

Modality-specific Transformers create embedding spaces that precisely describe the semantics of each modality and generally yield good results on a large variety of tasks. However, fusing the information from two different modalities is not trivial because the spaces created by both networks are not the same and, more importantly, are not aligned. Thus, because the two vectors lie in different spaces, simply concatenating the vectors of each modality does not create a meaningful representation. An explicit alignment of the two modalities is needed to fuse them in order to perform multimodal tasks. This alignment can be done early in the processing, at the feature level, creating a joint representation of the multimodal couple or later in the processing, at the scoring/decision level. Early fusion enables more interactions between the two modalities and creates a more fine-grained representation of the multimodal sample as a whole, but reduces monomodal processing and collapses the information into a single representation.

9.1.1 Cross-Modal Attention

In place of previous fusion methods [127] such as multimodal auto-encoder [242] or canonical correlation analysis [12], the attention layers offers a built-in fusion feature to Transformers. Following the duality between early and late fusion, multimodal Transformers are categorized as single stream [329, 197, 63] or dual stream [338, 211]. In the former, both modalities are concatenated to create a single sequence used as input of the model and processed jointly in a single Transformer through self-attention layers, maximizing the interaction between the two modalities. In the latter, they are extensively processed individually in separate modality-specific Transformers (thus using monomodal self-attention layers) before being fused and processed by a cross-modal Transformer that leverages cross-attention layers. As introduced with the decoder stack in Part I, a cross-attention layer can be defined by using key and value vectors from an external sequence to embed the information from this sequence into the sequence from which query vectors are extracted. Thus, a cross-attention layer can be leveraged to fuse information from two modalities by merging the information of the modality extracted using key and value into the information contained in the modality used as query. This implies a direction in the cross-attention, either from language to vision or from vision to language. In practice, a cross-modal attention layer is composed of two unidirectional cross-attention layers to model both directions. A representation of attention layers in single and dual stream

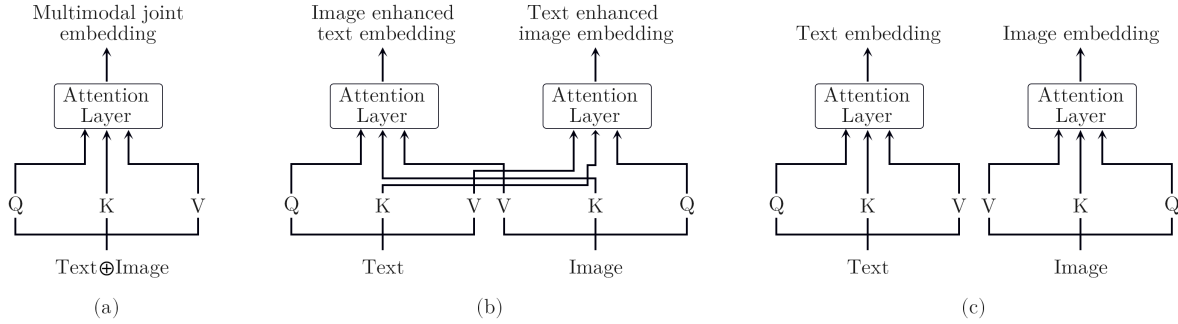


Figure 9.1 – Representations of the attention layers of different architectures for multimodal Transformers: (a) single stream, (b) dual stream, and (c) dual encoder. Note that dual stream Transformers also use monomodal self-attention layers interleaved with cross-modal attention layers that are not represented for the sake of simplicity.

multimodal Transformers is given in Figure 9.1.

Most vision-language Transformers use a pre-trained BERT [88] (along with its tokenizer) as the text encoder. For the visual input, they use the features of a pre-trained object detector such as Faster R-CNN [276]. These networks extract possible Regions of Interest (RoI) using a CNN representation fed to a region proposal network. The number of these regions is then reduced using non-maximum suppression, that is, merging multiple overlapping regions to the one with the highest confidence. These regions are then further processed to extract visual features associated with objects in the image. Thus, the image is represented by the features of its objects rather than the embedding of all of its patches as in usual visual Transformers [91].

9.1.2 Pre-training

The alignment between the textual and visual modality is done using parallel data, that is, an aligned couple of text and image where the text is related to the image, for example, its caption. The most intuitive objective to train alignment is Image Text Matching (ITM). Half of the time, the caption of the image is replaced with one randomly sampled in the dataset and the model is trained to predict if the input is a pristine couple or a randomly sampled one. The model thus has to learn to assess if an image and a text are aligned using the multimodal representation of the couple. Other specific tasks can be used such as Visual Question Answering (VQA) [123], where the model is asked to answer a question about the image, Natural Language for Visual Reasoning (NLVR) [330], where the model

is tasked to assert if a statement about the image is true, Visual Commonsense Reasoning (VCR) [406], which is similar to VQA with an additional layer of reasoning to answer the question or Visual Referring Expression (VRE) [171], querying the model for a fine-grained matching between a word of the text and a region of the image.

In addition to specifically cross-modal training objectives, vision-language Transformers are also trained with the objectives of text and vision-only Transformers. The MLM objective from BERT is used to train the text encoder. The main difference is that, since these models are multimodal, they can rely on the visual modality to resolve ambiguity (multiple words are likely in a given context, and the image may allow to select the correct one). Similarly, in the task of masked object prediction, an input object in the image (i.e., a token from the R-CNN) is masked rather than words in text, and the model is trained to predict either the features of the object (KL divergence/L2 loss with the feature of the R-CNN) or its label. Again, even if it is originally a monomodal task, the text associated with the image can be used to get clues (or even the exact answer), enhancing the cross-modal alignment. Similarly to monomodal Transformers, multimodal ones are pre-trained on large scale generic datasets such as Visual Genome [181] and Conceptual Captions (3M [310] and 12M [57]), as well as smaller datasets such as MS-COCO [206] and Flickr30k [394], and are then fine-tuned on task-specific datasets such as VQA v2 [123] and GQA [150].

9.2 Dual Encoder

9.2.1 Fast Retrieval

Even dual stream networks, that only model cross-modal interactions in the later stage of the processing, rely on heavy attention layers to fuse the information from both modalities. One of the shortcomings of such an approach is the scalability for certain tasks. For example, for cross-modal retrieval, this architecture requires computing the cross-attention between the query and every element in the database, which is very costly and becomes intractable when the database grows. One way to solve this problem is to use dual encoder models [270, 300], that is, separate encoders computing the representation of each modality, as illustrated in Figure 9.1. Rather than computing the joint representation of an image/caption couple, they project the image and the text separately in a common embedding space where the similarity between the two can be computed by a dot product as

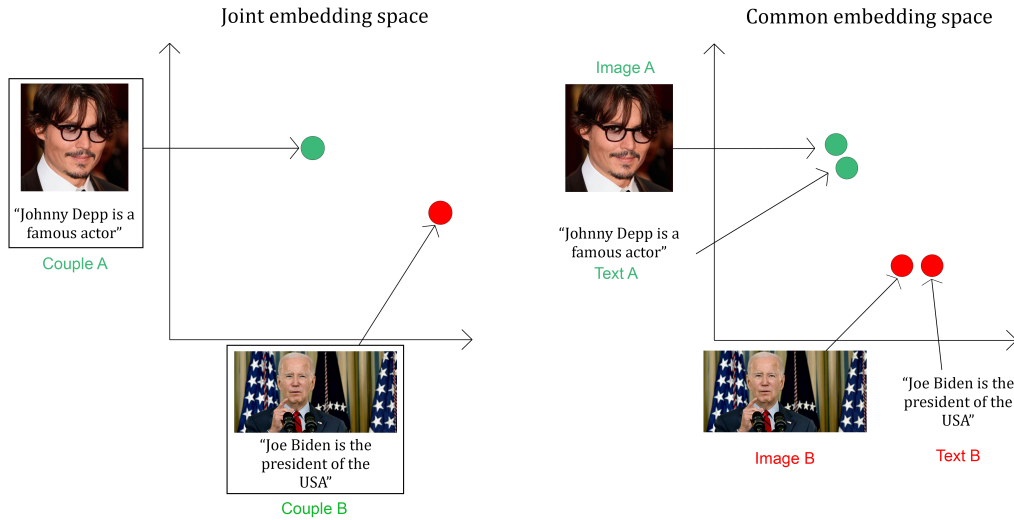


Figure 9.2 – Representation of a joint (left) and a common (right) embedding space. An image and a corresponding text are projected together in the joint embedding space, where the couple can be compared to another couple, whereas modalities are projected independently in the common space and can be compared to either the same modality or the other modality.

illustrated in Figure 9.2. This makes these methods *indexable* because the representations of the elements in the database can be pre-computed and only the representation of the query needs to be computed at inference time, allowing to use fast approximate nearest neighbor search (such as product quantization [160], inverted indexes [323], hierarchical clustering [237] or local sensitive hashing [81]) that allows to scale retrieval to billions of images. Such a light fusion between the two modalities does not enable complex cross-modal tasks such as VQA, thus these models are trained for their downstream task: determining if an image and a caption belong together. Although this is similar to the ITM objective previously described, the training is usually performed through contrastive learning.

9.2.2 Contrastive Learning

Contrastive learning is a representation learning paradigm which is particularly popular in computer vision and works by comparing different inputs. The objective is to create a representation space where "similar" inputs are close and "dissimilar" ones are far, with respect to a simple measure in the embedding space, such as the euclidean distance or the

cosine similarity. Thus, for a given input, the *anchor*, other inputs are defined as *positives* if they are similar (i.e., they share a property that the representation should be invariant to) and the others are *negatives* (i.e., the representation should be sensitive to the differences between them). In the case of images, the model is tasked to make the representation of an image closer to the representation of a transformation of this image than to the representation of any other image. Diverse transformations can be applied, ranging from color jittering to image flipping. In practice, the model will learn a representation that is invariant to these transformations and variant to the (small) differences between the represented image and others images. By doing so, it makes the representation robust to such transformations. Thus, in general, the transformations should not modify the semantics of the image so the model focuses on encoding those.

The simplest form of contrastive learning (with only one positive and one negative), the triplet loss, was originally introduced to build face representations robust to different poses and angles [299]. Given the anchor picture x of someone, a positive sample x^+ of the same person and a negative one of another person x^- are sampled. The encoder f is then trained to maximize the similarity of the positive and the anchor over the one of the anchor and the negative:

$$\mathcal{L}_{\text{triplet}}(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-) = \sum_{\mathbf{x} \in \mathcal{X}} \max\left(0, \|f(\mathbf{x}) - f(\mathbf{x}^+)\|_2^2 - \|f(\mathbf{x}) - f(\mathbf{x}^-)\|_2^2 + \epsilon\right) \quad (9.1)$$

with ϵ a margin parameter enforced between positive and negative pairs. This loss can be generalized to multiple negative samples, resulting in the N-pair loss [324]. InfoNCE loss [324, 248] extends this definition by using every other element in the mini-batch as negatives. It can be seen as a standard cross-entropy loss on the encoder representation similarities of elements within a batch B using the transformation of the anchor $T(x)$ as target label:

$$\mathcal{L}_{\text{InfoNCE}} = - \sum_{x \in B} \log \frac{\exp(f(x) \cdot f(T(x))/\tau)}{\sum_{k \in B} \exp(f(x) \cdot f(k)/\tau)}. \quad (9.2)$$

Hence, the learning objective is to maximize the softmax similarity of the positive $T(x)$ with the anchor x , that is to maximize the similarity of the positive and minimize the one of the negatives. The softmax effectively focuses on the highest similarities within the batch, so hard negatives (negatives that are very similar to the anchor) within the batch are required to learn a good representation space. Hard negatives can be obtained either through explicit mining or very large batch size [115, 132, 299, 221, 363]. The temperature

of the softmax parameter τ defines the hard negatives strength [363]. Low temperature makes the distribution sharper, increasing the penalty of negatives that are very similar to the anchor, whereas a higher value flattens the distribution, making it less sensible to false negatives. It defines a trade-off between the discriminativeness of the representation and robustness to noise in the dataset. This definition can be extended for a set of positives $P(x)$ for the anchor x , averaging the loss over every positive:

$$-\sum_{x \in B} \sum_{p \in P(x)} \frac{1}{|P(x)|} \log \frac{\exp(f(x) \cdot f(p)/\tau)}{\sum_{k \in B} \exp(f(x) \cdot f(k)/\tau)}. \quad (9.3)$$

Note that, although contrastive learning is mostly used in an unsupervised setting, label information in the contrastive objective has been shown to improve the results and creates representations more generalizable, at the cost of dataset labeling [174]. The label information is added in the contrastive objective, by defining the set of positives as elements of the batch sharing the same class as the anchor [174].

For multimodal data, two encoders are trained: one for images f^I and one for text f^T . For a positive couple of an image and its associated text (e.g., a caption) (i_c, x_c) , negatives are defined for both cross-modal direction (text-to-image and image-to-text) by using images \mathcal{I} and texts \mathcal{T} of the batch as negatives in Equation 9.3:

$$\mathcal{L}_{\text{CLIP}} = \underbrace{\log \frac{e^{\frac{f^T(x_c) \cdot f^I(i_c)}{\tau}}}{\sum_{x \in \mathcal{T}} e^{\frac{f^T(x) \cdot f^I(i_c)}{\tau}}}}_{\text{image-to-text}} + \underbrace{\log \frac{e^{\frac{f^T(x_c) \cdot f^I(i_c)}{\tau}}}{\sum_{i \in \mathcal{I}} e^{\frac{f^T(x_c) \cdot f^I(i)}{\tau}}}}_{\text{text-to-image}}. \quad (9.4)$$

An illustration of the cross-modal contrastive objective can be found in Figure 9.3. This objective effectively trains the encoders to produce embeddings for a couple that are the closest in the batch. After being trained on a dataset of 400 millions text-image pairs with a contrastive objective, CLIP [270] showed to be competitive with state-of-the-art on a large variety of tasks, including zero-shot classification and retrieval.

9.3 Fast Dual Encoder and Slow Cross-Encoder

Contrastive learning has become the most popular cross-modal retrieval pre-training strategy, enabling zero-shot capacities by scaling to billions of noisy text-image pairs directly scraped from the web. Such results catalyzed research efforts on these objectives, monopolizing attention to the detriment of cross-modal generative objectives such as image

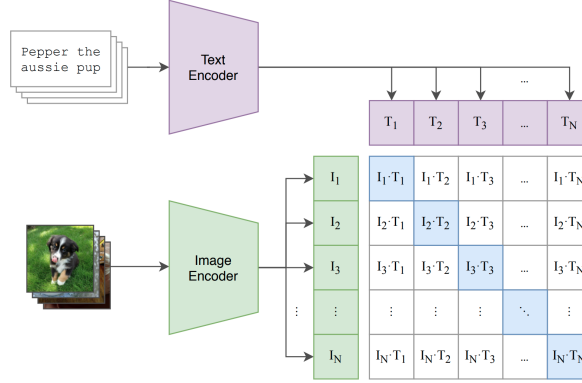


Figure 9.3 – Illustration of the cross-modal contrastive objective, taken from the CLIP paper [270]. The model is trained to maximize the diagonal and minimize other elements in rows/columns.

captioning. This is unfortunate, as a recent study shows that visual encoders trained with the generative objective are competitive with the ones trained using contrastive loss when used on downstream classification tasks and even surpass it for tasks that require fine-grained modeling of the language [348]. This granularity is needed for misinformation detection, where the language is often ambiguous and the detection relies on subtle clues. Besides, image captioning enables cross-modal retrieval. Indeed, an image captioning objective (introduced in Section 10.2) is a good pretext task for cross-modal retrieval and can replace a more traditional image-text matching objective [227]. The image captioning model is trained to compute the likelihood of a caption given an image, and so, given a text query (resp. image query), its likelihood to every image in the database (resp. every text) can be computed to get the top matching elements.

However, as described in the previous section, this retrieval process does not scale well. Image captioning models, such as BLIP [196] and LEMON [149], require computing the cross-attention between the query and every element in the database, which is not feasible at query time for large databases. Dual encoders are thus preferred because they allow to precompute the representations of all the elements in the database and enable fast retrieval on very large databases. Yet, dual encoder retrievers are known to perform worse than cross-encoders (Transformers relying on cross-modal attention for the fusion) [48, 137, 227]. A solution is to first filter the database with a fast dual encoder model to get a subset of the database that is close to the query, and then use a slow model based on the cross-attention on this small subset, to refine the ranking and get the best accuracy

even at a large scale [227]. Interestingly, the GCN framework introduced in Chapter 6 can be used to train **both the slow and fast models at the same time**. The generator requires cross-modal attention to be able to produce a correct caption for the image, but the discriminator that is used to score the produced text exhibits no such restriction and can be a dual encoder. Actually, in adversarial approaches, discriminators are often too powerful for the generator to learn properly due to sparse rewards and vanishing gradient [18]. This suggests that using models with less capacity as discriminators would be beneficial for the generator. Besides, jointly training both networks would be beneficial for downstream retrieval performance, because the slow network would be trained on the distribution of captions highly scored by the dual encoder, which is the distribution of captions that are likely to be retrieved by the fast model and that it will have to re-rank during inference. Finally, the difference in retrieval performance between dual encoder and cross-attention encoders might be due to the difference in the training objective more than the architecture itself and a proper distillation of the slow model, such as GCN, might close the gap [226].

The main difference with monomodal GCN is that, besides generating undetectable texts, the generator needs to produce a caption corresponding to the image. Hence, in addition to being trained to detect generated samples, the discriminator should be trained on the image-text matching task. The resulting rewards thus guide the generator to produce captions that are both undetectable and correctly describe the image. This can be achieved by training a MLP to classify the concatenation of the output representations of a dual encoder as generated or real.

The main issue of this approach is that it relies on the original multimodal space learned by the dual encoder and will not learn any new relationship. For example, if the dual encoder does not associate the breed of a bear with a picture of it, the generator will never learn to generate the actual breed name. Jointly training the generator and the dual encoder allows them to agree on the name of some objects. Yet, these new relations are not grounded in human language and the models can start using wrong terminology (e.g., replacing an object name with a color) or even create new words. This causes the generator to drift further and further from the human language [355, 179]. As a preliminary attempt to apply GCN to multimodal data, we study the setup where the **generator is trained using a pre-trained fixed cross-modal retriever**. Specifically, we explore how ground truth captions can be used to ground the learning to the human distribution.

DISTINCTIVE IMAGE CAPTIONING: LEVERAGING GROUND TRUTH CAPTIONS IN CLIP GUIDED REINFORCEMENT LEARNING

Contents

10.1 Introduction	128
10.2 Image Captioning	130
10.3 Method	133
10.3.1 Preventing Reward Hacking Using a Discriminator	133
10.3.2 Beyond a Single Baseline: The Bidirectional Contrastive Reward	134
10.3.3 Reward-Weighted Teacher Forcing	135
10.4 Experiments	136
10.4.1 Experimental Setting	136
10.4.2 Model Variants	137
10.4.3 Results	137
10.5 Conclusion	139

Image captioning is a competitive objective to create a cross-modal alignment [348, 227]. However, training image captioning using the standard teacher forcing results in very generic samples. This is damaging for the distinctiveness and thus for the downstream retrieval performances. Recent studies show that pre-trained retrieval models can be used to train the model to generate distinctive captions. In this chapter, we explore how ground truth captions can be used in this setup.

10.1 Introduction

Image captioning is the task of generating a description of the semantics of an image in natural language. One major challenge in this domain is to generate **distinctive** captions, that is, a description that allows to distinguish between the input image and other (similar) ones. For instance, "One person is standing" can be considered as a correct caption for several images showing someone: it is a correct sentence that fundamentally describes such images, yet it is not describing specifically a given image more than another. In contrast to generic ones, distinctive captions are more informative and descriptive. This is an expected property for retrieval applications, by indexing images using an appropriate textual representation, or to provide further details to people with vision impairment.

Captions in standard datasets [206, 310, 181, 300] only describe the most salient objects in the image, that are often common to many images. Thus, captioning models trained to match Ground Truth (GT) captions tend to generate overly generic captions and often produce the exact same caption for different images that share the same global semantics [80, 79, 365, 373, 65, 145]. The reason is that an easy way to optimize usual image captioning metrics based on word matching is to generate words that are common across training samples, and not to generate very specific words that are present in very few captions.

Since the goal of distinctive image captioning is to generate a caption able to distinguish an image from the others, one solution is to use a cross-modal retriever and to measure if the generated text allows to retrieve the target image among others [79, 216]. A language model can be trained to generate texts that optimize the retrieval score using reinforcement learning by learning from generated sequences that yield high scores. Recently, advances in cross-modal retrieval models enabled the use of fixed pre-trained models such as CLIP [270, 300] to guide the generator towards distinctive captions [65, 399, 411]. Using a fixed cross-



Figure 10.1 – Examples of images with an overly generic ground truth caption, a caption generated by a model without regularization (leading to reward hacking), and the caption generated by our approach (well written and distinctive).

modal retriever reduces the risk of the generator and the retriever cooperatively converging towards something closer to a hash function rather than to natural language. However, since the retriever has not been trained to evaluate the quality of the input text, but only its relevance to the image, it may assign a very high similarity score to ill-formed sequences and the LM will ultimately produce non-readable captions. A regularization of the generated sequences is thus still needed to avoid drifting too much from the natural language.

In this work, we propose a training method taking advantage of GT captions to optimize the trade-off between the distinctiveness and the writing quality of generated captions, illustrated in Figure 10.1. The use of cross-modal retrieval models in RL frees from the need for target reference captions, because the score of the produced sequence is computed by its similarity with the image rather than comparing it to a reference sequence. However, we argue that they can still be useful in this setup. First, they can be used to train a simple MLP to distinguish between real and generated samples. This discriminator can replace manually defined regularization criteria from other approaches that leverage pre-trained CLIP models. This results in a GAN [121] environment, where the discriminator and the generator improve together. Second, GT can further be used as candidate baselines in our proposed contrastive reward that uses the strongest baseline in a batch to reduce the

variance of the gradient estimation. Finally, they can be treated as generated sequences in the RL paradigm, resulting in a teacher forcing objective weighted by the similarity score of the caption to the image, thus promoting the most descriptive captions among these. This allows to learn to generate more distinctive captions using only GT captions. Coupling this objective with the more traditional RL one computed on samples generated by the LM allows to perform exploration while having a learning signal grounded to the human distribution that shares the same objective.

Background on training distinctive image captioning models is first introduced in Section 10.2. Then, we present our proposed approach by introducing 1) the use of the discriminator, 2) the contrastive rewards and 3) the use of ground truth as additional trajectories for the RL objective in Section 10.3. Finally, we compare the results of the approach to a model trained following [65] and provide some insights on our different contributions through different variations of the training in Section 10.4.

10.2 Image Captioning

Teacher Forcing. A basic approach to train an image captioning model is to train a LM to produce the caption x^{gt} while being conditioned to the input image i using teacher forcing. The image can be seen as additional previous tokens used as context, resulting in a very similar loss: $L_{\theta}(x^{gt}) = -\sum_{t=1}^T \log p_{\theta}(x_t^{gt} | x_{1:t-1}^{gt}, i)$.

Image captioning, when trained through TF, suffers from the same issues as any text generation task previously introduced. Firstly, the exposure bias [274] induced by the mismatch between the training and the generation process. The model is never exposed to its mistakes during training but will suffer from error accumulation at test time. Secondly, TF only considers one target sequence, whereas many different sequences can represent the same semantic content and be valid targets. Finally, the loss is defined at the token level, while the quality of a sample is defined at the finished sequences level.

Reinforcement Learning. Again, one way to overcome the limitations of TF is to directly optimize a sequence-level evaluation metric through RL [398, 274]. In the context of image captioning, the metric commonly optimized [277, 364, 198, 407, 149] is CIDEr [354]. As BLEU and ROUGE, this metric is computed at the sequence level by comparing sampled sequences to GT references. Thus, it is non-differentiable and is optimized through the REINFORCE algorithm [380]. As with teacher forcing, the loss is the same than

conditioning the generation on a textual context: using a baseline b , the gradient of the loss of a caption x of a generator parameterized by θ that obtains a reward $r(x)$ is $\nabla_{\theta} L_{\theta}(x) = -(r(x) - b) \nabla_{\theta} \log p_{\theta}(x)$.

Self-Critical Sequence Training (SCST) [277] is the most widely used method for training image captioning models. SCST is an extension of REINFORCE that uses the model itself as a baseline to normalize the rewards. During the training, the current model will be used to generate a sequence \hat{x} using test-time decoding method (e.g., Greedy Search (GS)) and uses its reward as a baseline ($b = r(\hat{x})$) for a sequence generated using a better decoding method, such as Beam Search (BS). The model probabilities of samples that are better than the current model will be increased and the ones of samples that are worse will be decreased. Hence, SCST optimizes a sequence evaluation metric as REINFORCE but strongly reduces the variance induced by sampling a full sequence while also avoiding learning a critic [334, 234] that estimates the expected future reward for a given sub-sequence.

Distinctive image captioning. Contrastive Learning for Image Captioning [79] introduces the distinctiveness property of image captioning models. The generator is trained using the log-ratio of probabilities of the model with respect to a reference baseline model on positive and negative pairs created by randomly swapping the positive pairs. The goal of the model is to assign higher probabilities to positive pairs (respectively, lower to negative ones) than the reference model. Our approach, on the other hand, does not directly work on sequences probabilities, that are optimized using reinforcement learning as a proxy. Leveraging a dual encoder model (CLIP) allows to compute scores for every pair in the batch and to consider much more couples than what would be tractable by evaluating the conditional probability of every sequence for this model. [65] use CLIP score in the SCST framework to train the model and fine-tune its text encoder to detect grammatical mistakes and prevent reward hacking. [411] improves over SCST by replacing the self-critical baseline with the CLIP similarity of the generated caption to a group of similar images and add a CIDEr reward to prevent the model from diverging. These rewards only focus on either text-to-image or image-to-text retrieval, whereas our approach considers a whole batch of similar captions and images, thus considering both directions. Moreover, rather than a fixed grammar network/CIDEr score, we use an evolving discriminator which adapts to the generator and prevents emerging behaviors that are not observable at the sequence level (e.g., low diversity that can only be measured from a set

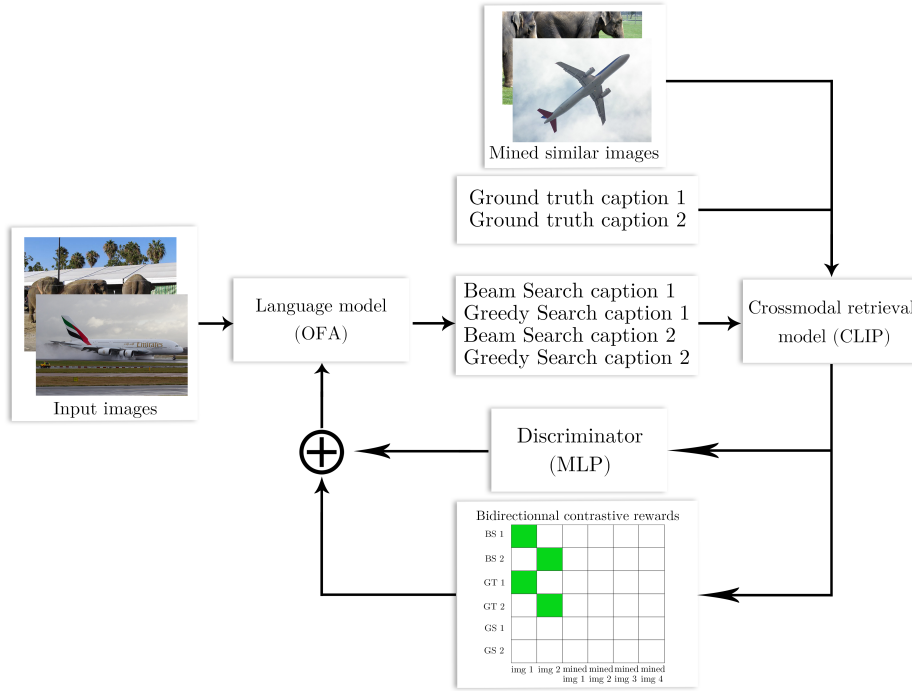


Figure 10.2 – Proposed captioning model learning overview. Generated and ground-truth captions, as well as input and mined similar images, are projected in the CLIP embedding space. Those representations are used to compute the reward composed of a discriminator score (Section 10.3.1) and a CLIP-based bidirectional contrastive similarity score (Section 10.3.2), for *beam search* and *ground-truth* samples (Section 10.3.3) (in green in the reward computation bloc).

of generated sequences). [145] identifies the limited vocabulary of a RL-trained model as a bottleneck for discriminativeness, preventing the model from using low-frequency words that are needed to correctly distinguish one image from another. This vocabulary collapse appears because only words sampled by the model obtain rewards, causing less frequent words to be less and less frequent [67]. [365] shows that using teacher forcing during the RL training limits the vocabulary collapse. Our proposed weighted teacher forcing combines the TF and RL objectives, by using ground truth captions that contain words that might not be sampled by the model.

10.3 Method

The proposed method extends the RL paradigm that uses the similarity score between a generated caption and its image from the cross-modal retriever CLIP as the reward.

$f^I(i)$ and $f^T(x)$ are respectively the CLIP embeddings of image i and text sequence x . For each image i in a batch, generated captions x^{bs} and x^{gs} are sampled from the LM using respectively beam search and greedy search decoding. We denote the ground truth caption by x^{gt} . The reward $r(x)$ is a trade-off between $r_{sim}(x)$, the similarity of $f^T(x)$ with $f^I(i)$, and $r_{regu}(x)$, a regularization based on the writing quality of the sample, controlled by a parameter α . The gradient is then given by:

$$\nabla_{\theta} L_{\theta}(x) = -r(x) \nabla_{\theta} \log p_{\theta}(x) \quad \text{with} \quad r(x) = \alpha r_{sim}(x) + (1 - \alpha) r_{regu}(x) \quad (10.1)$$

In the proposed learning scheme, depicted in Figure 10.2, the GT captions are leveraged (i) to train a simple discriminator D in the CLIP embedding space that discriminates between GT samples and the LM-generated ones, used as the regularization term (Section 10.3.1), (ii) as candidate baselines for the rewards of x (Section 10.3.2), and (iii) as additional training samples using the RL objective (Section 10.3.3).

10.3.1 Preventing Reward Hacking Using a Discriminator

During the exploration of the space by the policy (the LM), bad sequences with high rewards might be produced, for example by repeating important keywords for CLIP (reward hacking), or producing out-of-domain sequences that obtain near-random rewards. When ill-formed sequences get high scores, the model learns to reproduce them more and more, until the model fully collapses on this bad distribution (see Figure 10.1).

To prevent learning from such ill-formed solutions, previous approaches [65, 399, 411] use different criteria to regularize the training: detection of repetitions and grammatical errors, divergence from the distribution of the original LM, or CIDEr value. However, we argue that all of these criteria can be encapsulated in a single discriminator D trained to discriminate between GT samples and the LM-generated ones. A very simple MLP classifier taking as input the representations $f^T(x)$ and $f^I(i)$ computed by CLIP easily achieves a very high discrimination accuracy. The probability $D(x)$ of a sequence x to be from a human given by D can be used as $r_{regu}(x)$. It is worth noting that, contrary to the grammar head of [65], the discriminator is trained without fine-tuning the text encoder of

the CLIP model, preventing a mismatch between optimizing the retrieval model used for training and the one used at test-time. Although being less discussed in the literature, the discriminator is also useful to prevent the LM to degenerate in another situation: a well-written caption but insufficiently specific to the image (Figure 10.1). Such captions will obtain negative rewards because CLIP scores them poorly, but lowering their likelihood may lead the model to unlearn the grammar and correct sequence structure. While positive rewards attract the model directly toward the trajectory, negative rewards push it away in an unknown direction. It causes the model to produce random sequences that might make the model totally collapse. The weight of the discriminator in the reward should thus be large enough to prevent both unlearning well-written texts and learning from reward hacking samples. Compared to the grammar network of [65] or the CIDEr score of [411], our discriminator not only gives information at the sentence level, but more generally on the distribution of the generated texts and can adapt to emergent behaviors of the language model that could trick a fixed model.

10.3.2 Beyond a Single Baseline: The Bidirectional Contrastive Reward

Inspired by the contrastive loss used to train CLIP, defined for a given couple of (x_c, i_c) , a temperature parameter τ and a collection of negative texts \mathcal{T} and negative images \mathcal{I} , we derive a bidirectional contrastive reward, used as similarity reward r_{sim} in (10.1):

$$r_{bicont}(x_c) = \tau \left(\underbrace{\log \frac{e^{\frac{f^T(x_c) \cdot f^I(i_c)}{\tau}}}{\sum_{x \in \mathcal{T} \setminus x_c} e^{\frac{f^T(x) \cdot f^I(i_c)}{\tau}}}}_{\text{Image-to-text reward } r_{i2t}(x_c)} + \underbrace{\log \frac{e^{\frac{f^T(x_c) \cdot f^I(i_c)}{\tau}}}{\sum_{i \in \mathcal{I} \setminus i_c} e^{\frac{f^T(x_c) \cdot f^I(i)}{\tau}}}}_{\text{Text-to-image reward } r_{t2i}(x_c)} \right) \quad (10.2)$$

Please note that the reward more precisely corresponds to the definition of the decoupled contrastive loss of [392] that excludes the positive couple from the denominator. \mathcal{T} is composed of x^{bs} , x^{gs} , and x^{gt} for all images of the batch (see Figure 10.2). Although we initially chose to use the OFA [364] model as the generator to be able to generate both a baseline text and a baseline image, generating images was too expensive in practice for our resources. Hence, \mathcal{I} is composed of other images of the batch, as well as similar images mined in the dataset and added to the batch as additional images, following the setup of [411]. Note that the computation of the image representations $f^I(i)$ and the

mining of similar images is done only once before the training, and thus does not bring any computational overhead during training. As the representations $f^T(x^{gt})$ and $f^T(x^{bs})$ of GT and BS captions are already computed for the discriminator, computing the contrastive reward is thus inexpensive since it only consists of dot products.

Unlike previous approaches [65, 411], our contrastive reward considers both directions (by normalizing either by the similarity of the image with all the captions of the batch —image-to-text reward r_{i2t} —, or by the similarity of the caption with all the images in the batch —text-to-image reward r_{t2i} —), making sure that the caption is very descriptive of the image and this image only. Both parts of the reward can be rewritten as the similarity of the couple (x_c, i_c) minus the LogSumExp (LSE) of the similarities within the batch. With a small enough temperature parameter τ , LSE is an approximation of the max operator:

$$\begin{aligned} r_{i2t}(x_c) &= \tau \left(\log\left(e^{\frac{f^T(x_c) \cdot f^I(i_c)}{\tau}}\right) - \log\left(\sum_{x \in \mathcal{T} \setminus x_c} \frac{e^{f^T(x) \cdot f^I(i_c)}}{\tau}\right) \right) \\ &\approx f^T(x_c) \cdot f^I(i_c) - \max_{x \in \mathcal{T} \setminus x_c} \{f^T(x) \cdot f^I(i_c)\} \end{aligned} \quad (10.3)$$

This image-to-text reward can therefore be seen as the standard SCST where the baseline b is the hardest negative: the most similar caption to the image i_c among negative samples $\mathcal{T} \setminus x_c$. This is why x_c is excluded from the denominator, otherwise, the reward would always be negative or zero, even when x_c is the most similar caption to i_c among every caption in the batch (the goal of the model). Applied to the text-to-image reward, this approximation is almost equivalent to the G_{min} formulation in [411] that uses the similarity of the most similar image as the baseline. The proposed bidirectional contrastive reward thus seamlessly handles both cross-modal retrieval directions and selects the strongest baselines among a large batch, at a very low cost. The mean similarity in the batch is closer to the running average, often used as a baseline in traditional RL. However, early experiments showed that is not a strong enough baseline to prevent the model from diverging. The proposed reward results in a more conservative learning, only letting the model learn from very good sequences.

10.3.3 Reward-Weighted Teacher Forcing

Reinforcement learning scores trajectories (sequences of words in the context of text generation) and learn from those that scored well. Good sampled sequences are thus

required so that the model can learn from them. While this exploration process allows to find good solutions, it has a great variance and can lead to degenerate solutions (reward hacking).

GT captions can be considered a great source of relatively good solutions. We thus propose to use these captions as additional trajectories for the RL loss. If the reward is directly derived from the ground truth as in reference-based metrics (BLEU, ROUGE, CIDEr...), GT trajectories always obtain the upper-bound value of the reward metric. This reward is the same for every GT sequence, resulting in the standard teacher forcing objective with a learning rate multiplied by this constant. In our case, the cross-modal similarity score associated to the GT is not constant (some GT captions are closer to their images in the cross-modal space). The resulting loss is thus equivalent to the teacher forcing loss weighted by the reward $r(x^{gt})$. We refer to this objective as **Weighted Teacher Forcing (WTF)**.

The model still learns to reproduce human-written sequences but focuses more on the captions that are highly descriptive of their image, allowing to produce distinctive captions. Moreover, since these trajectories are written by humans, it strongly reduces the risk of reward hacking. Besides helping the model to stay close to the human distribution, it also enables to get back to it if the model reaches a pitfall (such as reward hacking or divergence), allowing the model to recover and start learning again. This is a very handful property since RL trainings are known to be unstable. Finally, since the rewards of every (image, text) pair are already computed, their associated contrastive rewards can be obtained at no additional cost. Applying (10.1) with the reward defined in (10.2) as r_{sim} and the discriminator as r_{regu} to both BS and GT captions, we end up with the following gradient for a given image:

$$\begin{aligned} \nabla_{\theta} L_{\theta}(x^{bs}, x^{gt}) = & - \left[\left(\alpha r_{bicont}(x^{bs}) + (1 - \alpha) D(x^{bs}) \right) \nabla_{\theta} \log p_{\theta}(x^{bs}) \right. \\ & \left. + \left(\alpha r_{bicont}(x^{gt}) + (1 - \alpha) D(x^{gt}) \right) \nabla_{\theta} \log p_{\theta}(x^{gt}) \right]. \end{aligned} \quad (10.4)$$

10.4 Experiments

10.4.1 Experimental Setting

We conduct our experiments on the MS COCO dataset [206] using the Karpathy splits [169]. We trained three variants of the proposed setup, that leverages a discriminator

D and uses the bidirectional contrastive reward: one using only GT trajectories (WTF), one using only generated BS trajectories (RL), and one using both (WTF-RL).

These models are compared to the training setup of [65], using the grammar network provided by the authors and the same weighting between the grammar and CLIP score reward (SCST-Grammar). All the models are trained, starting from the same checkpoint: the state-of-the-art captioning model OFA [364] in its tiny version trained using TF for 2 epochs, for which we also report the results as baseline (TF). They are trained for 5 epochs, using a learning rate of $1e - 6$ and α set to 0.94. The discriminator is first pre-trained to distinguish x^{gt} and x^{bs} from the original LM and is then trained throughout the LM training process.

Different metrics are used to compare different properties of generated samples. The Retrieval@k metric using the fixed pre-trained CLIP model (R@k) evaluates the discriminativeness of the generated caption. This metric is reported for $k \in \{1, 5, 10\}$ and for both text-to-image and image-to-text retrieval. Next, standard COCO captioning metrics that evaluate the writing quality are reported, including BLEU [426], ROUGE [255], CIDEr [354], METEOR [27] and SPICE [11]. The Self-BLEU [426] metric is also reported to measure the diversity in the generated samples.

10.4.2 Model Variants

Additional models are trained to study the influence of different elements in the training scheme. The gain brought by the bidirectional reward is studied by removing the text-to-image reward (RL-Unidirectional) from the contrastive reward. This reward is very similar to SCST with a baseline corresponding to the caption that has the highest similarity with the image in the batch, instead of using only the greedy search sample as the usual SCST. To evaluate the benefits of the discriminator, we also train a model using only the GS caption as the baseline (SCST). This last setup is the same as [65] (SCST-Grammar), but using a discriminator rather than the grammar network.

10.4.3 Results

Results are reported in Table 10.1. The first observable finding is that the WTF objective alone improves retrieval metrics over TF using only GT captions, without degrading the writing quality of the model. This shows that learning from the most distinctive human-written captions allows the LM to generate captions containing more important

details while staying close to the distribution of the GT captions. It is thus a better objective than TF to couple with the traditional RL one to act as an additional regularization and prevent vocabulary collapse, while allowing to recover if the model reaches a pitfall during RL training. The combination of the two objectives (WTF-RL) results in a model that achieves competitive retrieval results while maintaining high writing quality.

	T2I RETRIEVAL			I2T RETRIEVAL			WRITING QUALITY					DIVERSITY
	R@1 ↑	R@5 ↑	R@10 ↑	R@1 ↑	R@5 ↑	R@10 ↑	B4 ↑	R-L ↑	C ↑	M ↑	S ↑	Self-BLEU ↓
TF	17.14	39.06	51.14	23.98	49.72	61.94	32.73	55.43	109	27.19	20.69	70.49
WTF	20.52	44.58	57.66	29.32	56.72	69.08	32.9	55.57	110.2	27.46	21.26	61.45
WTF-RL	33.82	61.98	73.68	44.26	73.34	83.4	24.61	51.05	86.22	25.7	20.09	57.55
RL	35.24	62.9	75.3	46.68	75.28	84.66	21.59	49	76.06	25.01	19.21	58.01
RL - Unidirectional	31.52	58.34	71.04	45.86	74.4	83.4	21.45	48.14	78.53	24.75	19.83	62.3
SCST - Discriminator	34.72	62.46	74.22	51.38	79.08	87.54	16.54	44.62	46.21	24.31	18.46	68.88
SCST - Grammar	31.84	58.98	71.10	44.0	71.86	81.92	16.35	45.23	41.24	25.31	19.72	80.66

Table 10.1 – Captioning results on the MS COCO dataset (Karpathy splits). R@k correspond to the retrieval rate at k using the fixed CLIP model either using text queries (T2I) or image queries (I2T). Writing quality metrics includes BLEU@4 (B4), ROUGE-L (R-L), CIDEr (C), METEOR (M) and SPICE (S). The Self-BLEU metric measures the diversity.

Additionally, the computed representations of the GT allow to use a discriminator to replace the grammar network, yielding substantially lower Self-BLEU scores. Indeed, CLIP assigns great rewards to some stereotypical information structures such as "in the background" or "in the foreground". Since these sequences are grammatically correct, they are not penalized by the grammar network. The discriminator, however, adapts to the LM and learns that they are a useful clue to detect generated samples; so, it prevents the generator from generating these too often. As previously mentioned, the discriminator has a global view of generated texts and can detect some bad behaviors that are not visible at the sequence level and adapt to these emergent behaviors. Besides, using the discriminator also achieves a higher retrieval rate without degrading the writing quality.

When using only a unidirectional image-to-text reward, the text-to-image retrieval metrics significantly drop, illustrating that considering both retrieval directions in the reward is needed to produce a caption that is highly descriptive of this specific image only.

Finally, the image-to-text reward only using the GS baseline (SCST-Discriminator) results in higher retrieval results but lower writing quality. Selecting the strongest baseline in the batch (RL-Unidirectional) lowers the similarity part of the reward (r_{sim}), resulting, for a fixed α , in a higher weight of the discriminator. Although we can not conclude from

these results that using a stronger baseline yields a better retrieval/writing quality trade-off, it is expected to reduce the variance even more and prevent the model from committing too early. This can help preventing the exploitation of the reward model biases, especially in the early stage of the training, where the GS samples can be very weak. We recall that this does not bring computational overhead.

10.5 Conclusion

We studied how ground truth captions can be used in a reinforcement learning training that leverages a pre-trained cross-modal retrieval model in which they are no longer required. These captions can be used as **additional trajectories for the RL objective**, resulting in a weighted teacher forcing objective that allows to ground the exploration to the human distribution. This additional regularization could show very useful in a setup where the retrieval model is not fixed, by **forcing the model to use original captions and their vocabulary** while sharing the objective of generating distinctive captions. It also allows the model to recover from the inherent instabilities of RL training. They also can be used to **train a discriminator that will ground the exploration made by the policy to the human distribution** by favoring human-like generated samples. This signal serves as a regularization of the writing quality of the model, subsuming sequence-level criteria used in previous studies while preventing the emergence of bad behaviors that are not observable at the sequence level.

Finally, we leverage the fact that dual encoder models can compute the score of every pair in the batch at a low cost, and we use the definition of the decoupled contrastive loss to **select the strongest baseline in the batch**. This contrastive reward, in addition to being very similar to the original training setup of the reward model, **can be used in both cross-modal retrieval directions**, which is important to build truly distinctive captions.

Our findings pave the way for studies that try to also improve the CLIP model jointly with the captioning model. Starting from a strong pre-trained cross-modal retriever and strongly grounding the learning to ground truth captions might help to overcome the drifting inherent of the collaboration between the two models. To enable such extensions, the code of our approach is made [publicly available](#).

Considering image captioning models as language models with an additional conditioning input image, we conducted (non-reported) preliminary experiments on applying

cooperative generation to this multimodal setting. They show that **PPL-MCTS (Chapter 5)** can be used to guide an image captioning model with a CLIP model to produce captions that achieve higher CLIP scores than the unguided model. This indicates that a framework similar to generative cooperative networks can be applied to multimodal data. It also indicates that **Therapy (Chapter 7)** can be applied to **multimodal data** and study the behavior of a cross-modal retriever by extracting its most important keywords for an image using cooperatively generated captions yielding the highest scores.

CONCLUSION AND PERSPECTIVES

In this concluding chapter, we first summarize the main contributions of this thesis and introduce some perspectives of this work in Section 11.1. We then discuss the current state of generative models as well as their opportunities and risks, notably in the context of misinformation detection in Section 11.2.

11.1 Overcoming the Training Data Collection Challenge Through Data Generation

Generating training data In this thesis, we explored the use of synthetic, generated, textual data to train discriminative models when collecting training data is challenging. We first showed that the **original data can be replaced by data generated by a language model trained on the original data while maintaining comparable classification performance**, provided that a filtering is performed. Such data can also be used in addition to the original data, enhancing the performance of the resulting model, especially for simple and explainable models that benefit from reformulation such as Bag-of-Words models.

However, when a class is composed of only synthetic samples and another of only organic samples, the classifier leverages generation artifacts to make its decision. Hence, it focuses on discriminating the generated data rather than performing the intended discriminative task. Such clues allow the model to achieve high accuracy by learning features that are

not related to the target task (e.g., detecting misinformation) and will not hold in practice when used on organic samples. Because discriminators can detect generated samples, the GAN paradigm proposes to use them to train the generator towards undetectable samples. However, the discrete nature of text makes the training of generative adversarial networks challenging. Because the gradient from the discriminator can not be backpropagated to the generator, reinforcement learning is usually used. However, **because of the size of the search space, the training suffers from sparse rewards**, that is, the fact that only a very small proportion of all possible sequences actually obtain high rewards. Hence, it is difficult for the generator to produce samples obtaining good rewards, which are precisely the sequences it can learn from.

Cooperative generation To overcome this issue, we proposed to use **cooperative generation**, that is, using the score from the reward model directly during the generation process to guide the generator towards better samples. Existing cooperative methods suffered from a lack of a long-term view of the generation process, an essential component to generate coherent text and avoid committing to dead-ends. We explored the use of **Monte Carlo Tree Search, an algorithm that selects short-term action based on long-term results**, to guide the generation process. We showed that it can be used to generate texts that satisfy a constraint defined by an external classifier, such as a sentiment or a topic, thanks to its long-term view. We then leveraged cooperatively generated texts to train the generator and introduced **a novel formulation of GANs for the discrete setting that has theoretical convergence guarantees**. These are obtained by sampling from a distribution that is a combination of the generator and the discriminator distribution. Because sampling from this exact distribution is intractable, we sample from an arbitrary distribution and use importance sampling weights to unbiased the gradient estimation. However, sampling as close as possible to the target distribution is still necessary to limit the variance. Thus, we leverage cooperative generation to sample texts that are highly scored by both the generator and the discriminator to be closer to the target distribution. Using the MCTS yields the best results because it generates texts that are the closest to the target distribution (the mixture of the generator and the discriminator).

We also explored the use of cooperative generation to **generate texts that are representative of the distribution effectively learned by a classifier to improve its explainability**. This allows to study its behavior and find potential biases on the domain defined by the generator. Besides providing global explanations, this approach

allows to study the behavior of a model without requiring input data. This is particularly useful when there is no data available or it is not representative. Besides, the search of important feature is driven by the language model rather than the example, allowing a wider search.

Finally, we studied the **impact of the choice of the classifier architecture on the quality of the guidance and the computational complexity of the cooperative generation process**. While Transformers that leverage bidirectional attention are preferred for discriminative tasks, they are not adapted to cooperative generation because they require recomputing every hidden state at every generation step. We show that using a **unidirectional classifier results in a small drop in accuracy**, which reflects on the guidance quality. However, this is **balanced by the lower computational complexity**, which allows to increase the number of MCTS simulations to close the gap while still being faster than the bidirectional classifier. Surprisingly, generative discriminators, which seemed very interesting at first glance due to their ability to score the whole vocabulary at once, allowing wider search, yield poor results. Indeed, scoring every token at once comes at the price of a lower accuracy, which reflects on the guidance quality. Because the distribution of the language model already selects the few tokens that can fit the sequence, MCTS explores more in depth than in width. Thus, it does not benefit from the wide scoring ability of the generative discriminator while suffering from its less informative guiding signal.

Perspectives of cooperative generation Although we extensively studied cooperative generation in this thesis, there are still **many use cases to be explored**. The most straightforward application would be to leverage it to add the filtering part of Chapter 2 directly into the generation process. It would also be interesting to study the benefits of doing multiple generation/training loops for classification tasks, à la GAN. Furthermore, a key component of actual best language models such as GPT-4 [249] and Llama 2 [347] is **Reinforcement Learning From Human Feedback (RLHF)**. In this setup, the language model is trained to optimize a sequence-level reward using reinforcement learning, as in textual GANs, but the difference lies in the reward. Rather than the score of a discriminator that detects generated samples, it optimizes the score of a model trained to model human preferences. The training of the reward model is done using pairs of sequences ranked by human annotators and the model is tasked to output a higher score for the better sequence of the pair. The model is then trained to produce sequences that

maximize this reward and thus the alignment with human preferences rather than being indistinguishable from human sequences. This obviously requires more costly annotations than GANs but seems like a very effective way to align generative models with our needs and values reflected in these annotations. Although the reward model is kept fixed and can not benefit from the positive training loop of GANs (and so from our introduced GCN), **cooperative generation could still be useful in this paradigm**. As with the GAN discriminator, it could help the generator to produce better sequences and achieve higher rewards by guiding it with the reward model. And, in a similar fashion to GANs again, these better sequences could enhance the reinforcement learning training or be used for imitation learning as in SelfGAN [304]. Also, using the value network produced during the reinforcement learning training to guide the generator yields even better results due to its inherent ability to score partial sequences [207].

Use cases going beyond our studies have already been proposed, including constraining the generation using factual consistency [400], logical constraints [212] and unit tests for code generation [408]. Cooperative generation might also be useful in the field of adversarial examples, by using the discriminator signal to generate an example that is as close as possible to the decision boundary while being misclassified [122]. Please note that most of the presented use cases of cooperative generation require explicitly defining the external constraint, emphasizing its interest over tuning the language model. Indeed, even though LoRA [147] enabled the training and storing of different versions of a language model that generate texts with a specific property, making the original use case of PPL-MCTS obsolete, there are still many others for which it is still relevant. On the contrary, as introduced in Chapter 5, using PPL-MCTS on a tuned language model will yield better results. Besides, if the guiding model and the language model share the same base model, LoRA offers a way to reduce the memory footprint of cooperative generation [295]. Exploring other cooperative generation methods that make the most of the width exploration of generative discriminators is also a promising avenue. In a similar vein, classifier-free guidance have recently been proposed for language models [293]. It can be seen as a **special case of generative discriminator and can be combined with cooperative approaches in the same way**. Finally, although we focused on generating texts, it should be noted that our studies also apply to any discrete data, such as categorical inputs [30].

Cross-modal generation Even though cooperative approaches did not allow to generate a fully synthetic class of the dataset, improving generative models remains beneficial for

other types of data augmentation that use such models as the one presented in Chapter 2. Besides, generative objectives are a great pretext task to learn good representation spaces. As a preliminary work to integrate the visual modality in this cooperative environment, we explored the use of generative models to create a cross-modal alignment. To create the **most fine-grained alignment possible**, we focus on the **distinctive image captioning task**, that is, generating captions that are very descriptive of the input image and this image only. Specifically, we showed that **ground truth captions** can be leveraged in a reinforcement learning training using rewards from a cross-modal retrieval model where they are not needed, to ground the training to the original distribution. We showed that they can be used as **samples for the reinforcement learning objective**, resulting in a teacher forcing objective weighted by the reward. This objective is used to train the model to reproduce human samples while focusing on the most distinctive ones, matching the traditional RL objective. We further leverage the ground truth captions to **train a discriminator that serves as a regularization term to the generator** to further ground the generator to the human distribution. This grounding is a first step towards training both models jointly by limiting the inherent drifting from the human language due to the cooperation of the two models. Finally, we introduce a **contrastive reward, that consider every element in the batch as baselines for the reward**, letting the generator learn from the best sequences only. This contrastive reward, in addition to being very cheap to compute, **natively considers both cross-modal retrieval directions**, enabling to produce captions that are very descriptive of the input image and this image only. Given that image captioning models are language models with an additional conditioning input image, captions achieving high CLIP scores can be obtained by using PPL-MCTS to guide such models with a CLIP model. Thus, exploring a **multimodal framework similar to GCN and SelfGAN is a promising avenue of research**. It also indicates that **Therapy applies to multimodal models**, provided that one of the modalities is text.

11.2 Generative Models and Misinformation: A Double-Edged Sword

Growing capacity and accessibility Generative models have come a long way since ELIZA [378]. The quality of generated texts is increasing at an unbelievable pace and does not seem to slow down. Not so long ago, GPT-2 [269] public release was delayed

because it was considered "too dangerous to release"¹. Although there is no consensus on the question of public release [325, 405], it might sound funny given the capacity of today's public models. Better models are published daily (FreeWilly2 [5] was literally published two days after Llama 2 [347]), to the point where surveys are needed to keep up the pace [412]. Even though it is easy to get lost in this ever-growing number of models, it has never been easier to use them, thanks to libraries such as the [Hugging Face](#) ecosystem that lets anyone access the latest models, datasets and demos in a few lines of code. Swapping between different models is as easy as modifying a variable name. Even the training and inference of such models is getting easier with methods such as LoRA [147] and quantization [87, 86], and can even run on non-specialized hardware with low resources^{2 3}.

Opportunities of large language models Such powerful models are very capable and can be used as a helpful assistant for a large variety of tasks in our daily life, up to the point that it is expected to have a huge impact on the labor market⁴ [94]. Ideally, these models should facilitate the fact-checking job by pre-processing the huge amount of information available on the internet and flagging the most suspicious ones, while giving insightful clues for the human fact-checker to make the final decision. This is especially true for retrieval-augmented language models [130, 195, 42, 154], that will ground their generation process into known facts using the retrieved original source of the information or related information and can pass these sources to the fact-checker to speed up its verification. This is particularly helpful given how effective external pieces of evidence are against image repurposing [2]. Remind that this external knowledge can also take the form of knowledge graphs and that enhancing language models with structured knowledge bases is a promising avenue of research [250].

Besides, as highlighted in this thesis, strong generative models can be used to generate training data for other models. While generators are still not good enough to create a whole synthetic class, other data augmentation schemes leveraging generative models benefit from advances in those. For example, using synthetic data yielded very strong performances in image captioning [196], text-to-image generation [35] and code generation [284]. Other recent examples include the use of image captioning models to replace noisy captions from

1. <https://www.theverge.com/2019/11/7/20953040/openai-text-generation-ai-gpt-2-full-model-release-1-5b-parameters>

2. <https://github.com/ggerganov/ggml>

3. <https://github.com/huggingface/peft>

4. <https://www.goldmansachs.com/intelligence/pages/generative-ai-could-raise-global-gdp-by-7-percent.html>

the web instead of deleting them [243], and training a LLM using demonstrations [340] or instructions [146] generated from another strong LLM as a sort of distillation.

In addition to being a good objective for learning representations of different input modalities, generative tasks allow to query the model about the representation space to examine the knowledge it has learned. Asking diffusion models to generate an image of "an armchair in the shape of an avocado" gives insightful information on how these concepts are encoded in the embedding space. Cooperative generation adds this capacity to any discriminative model as illustrated in the introduced approach Therapy, at the cost of an inevitable additional noise.

Risks of large language models However, despite the numerous positive aspects of generative models, they also present significant risks. Indeed, humans are easily fooled by these models and cannot reliably detect synthetic texts [405, 71, 151], making the Turing test [349] obsolete. Although it is possible to enhance our detection capacity through appropriate training [92], a constant exposition to generated samples coupled to an increased performance will make the human detection of generated samples more and more difficult, increasing the risks and effectiveness of malicious use of these models such as automatic disinformation [119, 377]. Unfortunately, even with proper alignment training and safety controls to prevent the generation of harmful content, the models can still be attacked ("jailbroken") [429] and the risks of misuse are still present.

One might think that, while the fact that neural discriminators easily detect generated samples was a burden for our needs, it offers a solution to artificially generated disinformation [72, 25, 159]. However, even without reaching the generator optimum and thus achieving an indistinguishable distribution, as pointed out earlier, the discriminator is heavily specialized on the generation mode of the generator [302, 16, 151, 25, 325]. This means that any change in the generation process, such as the temperature or the sampling scheme makes the performance of the discriminator collapse, let alone a different model (especially if trained on a different domain). This is a major issue because, in addition to the growing number of LLM available, an attacker can simply retrain its model to fool the discriminator. Discriminating between the true distribution and multiple very similar ones is a really hard task. Even though discriminators exhibit some generalization capabilities to different generators, it seems that creating an all-in-one model able to detect every generator might be impossible [15, 267]. Besides, the good performance of discriminators in the GANs setting is due to a sufficient exposition to generated samples, which might

not be the case in the real world, especially for malicious actors [56]. Besides, as any neural classifier, the discriminators are sensitive to adversarial attacks [253], that create texts samples specifically built to exploit the discriminator weaknesses and be misclassified [122]. In our case, this means that generated texts can be tweaked to not be detected by a discriminator [383, 16]. Finally, better discriminators can be used to train better generative models, and cooperative generation presented in this thesis can be used to generate samples that are, by construction, not detected by a given discriminator. This results in a cat-and-mouse game where defenders need to quickly adapt to keep up with the attackers.

Considering all these factors, even though we managed to train a sufficiently good generic discriminator, the false positive rates would still be too high to be used in a real-world setting. Even with a nearly perfect detection rate, when applied to a real-world scale, many false positives will be found. When training the generator, detecting a real sample as generated is not really hurtful, but disqualifying honest competitors from photo contest⁵ or flunking a student who did not cheat⁶ is not acceptable. Yet, this paranoid climate is not surprising, given that generative models can be and are effectively used to cheat on exams [21]^{7 8} or win photography awards⁹. This is even more concerning given that these false positives can be biased towards a specific community [203]. Without further measures, detecting generated texts seems too unreliable [291, 352], to the point that even OpenAI themselves shut down their AI classifier "due to its low rate of accuracy"¹⁰ (26% true positives and 9% of false positive). Also, at the time of starting the thesis, the generation capacity of language models was still limited, let alone the capacity of image generation. Only very specialized networks, such as face generation networks, were able to generate acceptable samples. Apart from cats, dogs and faces, generated images were still clumsy. It was not a viable option for our approach, because the control over the generated samples was nonexistent. However, in less than a year after the initial release of Dall-E [273], the first model that enabled (convincing) zero-shot text-to-image generation, diffusion models [281] are getting better and better, offering large granularity on the control

5. <https://news.artnet.com/art-world/australian-photographer-disqualified-ai-generated-2337906>

6. <https://www.washingtonpost.com/technology/2023/05/18/texas-professor-threatened-fail-class-chatgpt-cheating/>

7. <https://www.theguardian.com/technology/2023/may/18/ai-cheating-teaching-chatgpt-students-college-university>

8. <https://www.nytimes.com/2023/01/16/technology/chatgpt-artificial-intelligence-universities.html>

9. <https://www.bbc.com/news/entertainment-arts-65296763>

10. <https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text>

of the generated images. It is now possible to generate an image indistinguishable for a human from a real one with only a small caption describing the desired image¹¹. The generation of other modalities, such as video and voice, while still lagging a bit behind, are making rapid progress and will certainly experience the same exponential progress and also raise several problematic use cases [319, 141, 97, 192, 362], notably impersonification and scamming¹². Convincing tools are already available for the general public^{13 14}.

Mitigating the risks with watermarking Watermarking can be part of the solution, either for images [105, 360, 425] or texts [175, 1, 413, 68, 183], but requires to be added **in** the model and not at decoding time [105] to make it costly for the attacker to remove it¹⁵. Watermarking works by modifying the output distribution of generators to add an invisible trace to the generated text/image, that will facilitate subsequent detection. However, it still requires exploration, for example to study resistance to attacks [176, 183, 291]. As part of this exploration, **our recent work [104] consolidates existing watermarking schemes for large generative language models in three different ways**. First, we introduce new statistical tests that offer robust theoretical guarantees, even at low false-positive rates, making the detection more trustworthy. Second, we measure the performance degradation induced by the watermark on practical NLP benchmarks, instead of metrics measuring the distortion to the original distribution. This is more informative of the utility of the model for downstream tasks, where the real interest of LLMs lies. Finally, we introduce parallel computation of multiple watermarks detection, turning the zero-bit watermarking (marked or not) to multi-bit watermarking (if watermarked, get the corresponding identifier). It allows the addition of a message to the watermarked text, such as the user that generated the text or the version of the model used. Although watermarking is a promising solution for the open release of generative models and big names of AI seem willing to add such watermarking to their generative models^{16 17}, malicious actors can still train their own models from scratch, heavily fine-tune a pre-trained model to

11. <https://www.nytimes.com/2023/04/08/business/media/ai-generated-images.html>

12. <https://consumer.ftc.gov/consumer-alerts/2023/03/scammers-use-ai-enhance-their-family-emergency-schemes>

13. <https://runwayml.com/>

14. <https://elevenlabs.io/>

15. <https://medium.com/@steinsfu/stable-diffusion-the-invisible-watermark-in-generated-images-2d68e2ab1241>

16. <https://www.theverge.com/2023/7/21/23802274/artificial-intelligence-meta-google-openai-white-house-security-safety>

17. <https://www.deepmind.com/blog/identifying-ai-generated-images-with-synthid>

remove the watermark or manually create misinformation. Even though the watermark can be useful to help detect generated content, we cannot assume that it will be added by everyone in every synthetic sample. Thus, a more viable use of watermarking might come from reversing the paradigm by adding the watermark to authentic samples, for example by adding the watermarking process directly within the camera. In this framework, the absence of a watermark indicates that the content might be generated. This implies that the content can only be trusted if it has a watermark from a known publisher. It aligns with the hypothesis that the Internet will soon be filled with generated content and we should start to assume that everything is synthetic unless proven otherwise. Furthermore, the watermark could be used to robustly link the image to a register of information on the image and its modifications such as [the C2PA](#) that “implements authentication technology to preserve trust in photojournalism”¹⁸. The link to the database would be directly added **in** the image and can not be removed, ensuring that the original context will always be found.

Fighting misinformation Fighting misinformation on social networks requires them to change the focus from maximizing user engagement to maximizing the quality of the information, deliberately or through government regulations [188]. It may take different forms, such as blocking propagation paths using propagation analysis and targeting the sources of fake news (influential users/bots). Blocking the misinformation before it is seen is crucial because even after detecting the misinformation, correcting it is a real challenge. Indeed, the debunking will not spread as much and as fast as the original false information [359]. Besides, it is really difficult to change the opinion of someone after he has been convinced by fake news [83]. Some solutions have been proposed, such as suggesting educational and personal recommendations of certified news and refutation of fake ones and highlighting more debunkers [358]. Yet, how to effectively present fact-checks is still an open question [20]. Social networks start to take action, for example, X (formerly Twitter) asks users to read the full article before resharing it¹⁹ to prevent spreading misleading titles. They also added "community notes", where every user can add information on a post, to give context or signal incorrect information²⁰. While this allows regular users to

18. <https://www.reutersagency.com/authenticity-poc>

19. <https://www.theverge.com/2020/9/25/21455635/twitter-read-before-you-tweet-article-prompt-rolling-out-globally-soon>

20. <https://help.twitter.com/en/using-twitter/community-notes>

take part in the fact-checking initiative, it is still far from sufficient to curb the problem²¹. Most importantly, it is essential to create practical tools that give the power of AI to fact-checkers²². Although these tools are not and might never be perfect to the point that they can be used to reliably automatically detect misinformation, they can certainly be used to speed up the process and help the fact-checker make the final decision. For example, by indexing very large cross-modal databases to retrieve insightful clues, highlighting possible problematic claims, detecting malicious networks of misinformation propagation, etc. When it comes to misinformation, artificial intelligence can be a double-edged sword, so it is important to make sure that we benefit from the best of the ever-improving AI²³ and not just suffer from the worst.

21. <https://www.poynter.org/fact-checking/2023/why-twitters-community-notes-feature-mostly-fails-to-combat-misinformation/>

22. <https://www.veraai.eu/home>

23. <https://time.com/6300942/ai-progress-charts/>

PART IV

APPENDICES



PPL-MCTS: CONSTRAINED TEXTUAL GENERATION THROUGH DISCRIMINATOR-GUIDED MCTS DECODING

A.1 Complementary Results

We tested three temperature values for each proposed method: 1.0, 1.1 and 1.2. As the temperature increases, the output distribution of the language model becomes more and more uniform. This means that high temperatures should result in high perplexities because the sampling deviates further from the original distribution.

For PPL-MCTS, we also studied the impact of c_{puct} by testing values 1.0, 3.0, 5.0 and 8.0 along with the different temperature values mentioned. c_{puct} defines the compromise between exploiting nodes that already have great scores and exploring less played but promising ones. A high c_{puct} encourages exploration. We remind that the repetition penalty I in Equation 5.2 has been set to 1.2 as defined in CTRL.

In Section 5.3.2 we only reported the results obtained with the set of parameter values yielding the best accuracy for each method and dataset. Hereafter, we report results with every tested set of parameters in Tables A.1, A.2 and A.3 for respectively the emotion, CLS and amazon_polarity datasets.

Unsurprisingly, the perplexity of methods which sample on the LM logits explodes when τ increases, without a noticeable gain in accuracy. Since the diversity is already high for low τ values, it seems to be better to keep the temperature low with these approaches. Note that the couple $c_{puct} = 3, \tau = 1.0$ for PPL-MCTS always leads to the best result. Using $c_{puct} = 8$ seems to yield slightly worse results, especially with a low temperature. However, the different parameters do not greatly affect the results of PPL-MCTS.

Generation method	Accuracy \uparrow	5 - Self-Bleu \downarrow	Oracle perplexity \downarrow
Beam sampling - Argmax $\tau = 1.0$	0,61	0,41	3,7
Beam sampling - Argmax $\tau = 1.1$	0,65	0,48	3,72
<i>Beam sampling - Argmax $\tau = 1.2$</i>	<i>0,67</i>	<i>0,48</i>	<i>3,88</i>
Beam sampling - First true $\tau = 1.0$	0,58	0,4	3,68
Beam sampling - First true $\tau = 1.1$	0,64	0,48	3,69
<i>Beam sampling - First true $\tau = 1.2$</i>	<i>0,66</i>	<i>0,49</i>	<i>3,85</i>
Beam sampling - Sampling $\tau = 1.0$	0,59	0,41	3,69
Beam sampling - Sampling $\tau = 1.1$	0,64	0,49	3,69
<i>Beam sampling - Sampling $\tau = 1.2$</i>	<i>0,66</i>	<i>0,48</i>	<i>3,88</i>
CC-LM - Greedy Search	0,51	0,1	17
<i>CC-LM - Sampling $\tau = 1.0$</i>	<i>0,52</i>	<i>0,13</i>	<i>11,1</i>
CC-LM - Sampling $\tau = 1.1$	0,51	0,1	15,8
CC-LM - Sampling $\tau = 1.2$	0,47	0,08	31,4
<i>CC-LM - Classloss - Greedy Search</i>	<i>0,89</i>	<i>0,65</i>	<i>3,72</i>
CC-LM - Classloss - Sampling $\tau = 1.0$	0,83	0,11	19,6
CC-LM - Classloss - Sampling $\tau = 1.1$	0,79	0,07	33,2
CC-LM - Classloss - Sampling $\tau = 1.2$	0,79	0,05	64,8
<i>Sampling - Argmax $\tau = 1.0$</i>	<i>0,87</i>	<i>0,13</i>	<i>11,7</i>
Sampling - Argmax $\tau = 1.1$	0,86	0,1	19,6
Sampling - Argmax $\tau = 1.2$	0,86	0,07	47,5
<i>Sampling - First true $\tau = 1.0$</i>	<i>0,82</i>	<i>0,13</i>	<i>10,4</i>
Sampling - First true $\tau = 1.1$	0,81	0,11	16,2
Sampling - First true $\tau = 1.2$	0,77	0,09	33,2
<i>Sampling - Sampling $\tau = 1.0$</i>	<i>0,81</i>	<i>0,13</i>	<i>10,4</i>
Sampling - Sampling $\tau = 1.1$	0,8	0,11	15
Sampling - Sampling $\tau = 1.2$	0,79	0,08	25,7
PPL-MCTS - $c_{puct} = 1.0, \tau = 1.0$	0,83	0,37	4,81
PPL-MCTS - $c_{puct} = 1.0, \tau = 1.1$	0,8	0,36	4,9
PPL-MCTS - $c_{puct} = 1.0, \tau = 1.2$	0,82	0,33	4,97
<i>PPL-MCTS - $c_{puct} = 3.0, \tau = 1.0$</i>	<i>0,84</i>	<i>0,37</i>	<i>4,82</i>
PPL-MCTS - $c_{puct} = 3.0, \tau = 1.1$	0,82	0,35	4,85
PPL-MCTS - $c_{puct} = 3.0, \tau = 1.2$	0,84	0,35	4,9
PPL-MCTS - $c_{puct} = 5.0, \tau = 1.0$	0,84	0,38	4,74
PPL-MCTS - $c_{puct} = 5.0, \tau = 1.1$	0,84	0,34	4,79
PPL-MCTS - $c_{puct} = 5.0, \tau = 1.2$	0,84	0,33	4,88
PPL-MCTS - $c_{puct} = 8.0, \tau = 1.0$	0,81	0,38	4,71
PPL-MCTS - $c_{puct} = 8.0, \tau = 1.1$	0,81	0,37	4,72
PPL-MCTS - $c_{puct} = 8.0, \tau = 1.2$	0,82	0,35	4,79

Table A.1 – Results for every tested set of parameters on the proposed methods; emotion dataset. Results reported in the main body are in italic.

Generation method	Accuracy \uparrow	5 - Self-Bleu \downarrow	Oracle perplexity \downarrow
Beam sampling - Argmax $\tau = 1.0$	0,87	0,71	3,85
<i>Beam sampling - Argmax $\tau = 1.1$</i>	<i>0,88</i>	<i>0,67</i>	<i>3,91</i>
Beam sampling - Argmax $\tau = 1.2$	0,88	0,63	4,12
<i>Beam sampling - First true $\tau = 1.0$</i>	<i>0,85</i>	<i>0,71</i>	<i>3,8</i>
Beam sampling - First true $\tau = 1.1$	0,84	0,68	3,87
Beam sampling - First true $\tau = 1.2$	0,85	0,63	4,07
Beam sampling - Sampling $\tau = 1.0$	0,74	0,71	3,82
Beam sampling - Sampling $\tau = 1.1$	0,72	0,68	3,89
<i>Beam sampling - Sampling $\tau = 1.2$</i>	<i>0,76</i>	<i>0,63</i>	<i>4,07</i>
CC-LM - Greedy Search	0,59	0,57	2,51
CC-LM - Sampling $\tau = 1.0$	0,62	0,15	12,3
CC-LM - Sampling $\tau = 1.1$	0,63	0,09	18,7
<i>CC-LM - Sampling $\tau = 1.2$</i>	<i>0,66</i>	<i>0,06</i>	<i>31,5</i>
CC-LM - Classloss - Greedy Search	0,8	0,59	2,77
CC-LM - Classloss - Sampling $\tau = 1.0$	0,85	0,13	17
CC-LM - Classloss - Sampling $\tau = 1.1$	0,87	0,07	28
<i>CC-LM - Classloss - Sampling $\tau = 1.2$</i>	<i>0,89</i>	<i>0,04</i>	<i>50,6</i>
<i>Sampling - Argmax $\tau = 1.0$</i>	<i>0,92</i>	<i>0,12</i>	<i>14,3</i>
Sampling - Argmax $\tau = 1.1$	0,92	0,08	20,7
Sampling - Argmax $\tau = 1.2$	0,92	0,05	33,6
<i>Sampling - First true $\tau = 1.0$</i>	<i>0,87</i>	<i>0,14</i>	<i>13</i>
Sampling - First true $\tau = 1.1$	0,86	0,1	19,1
Sampling - First true $\tau = 1.2$	0,86	0,06	33,1
Sampling - Sampling $\tau = 1.0$	0,77	0,14	12,9
Sampling - Sampling $\tau = 1.1$	0,78	0,09	18,8
<i>Sampling - Sampling $\tau = 1.2$</i>	<i>0,81</i>	<i>0,06</i>	<i>31,8</i>
PPL-MCTS - $c_{puct} = 1.0, \tau = 1.0$	0,88	0,54	4,98
PPL-MCTS - $c_{puct} = 1.0, \tau = 1.1$	0,87	0,53	5
PPL-MCTS - $c_{puct} = 1.0, \tau = 1.2$	0,87	0,53	5,02
<i>PPL-MCTS - $c_{puct} = 3.0, \tau = 1.0$</i>	<i>0,89</i>	<i>0,54</i>	<i>4,98</i>
PPL-MCTS - $c_{puct} = 3.0, \tau = 1.1$	0,89	0,54	4,81
PPL-MCTS - $c_{puct} = 3.0, \tau = 1.2$	0,89	0,54	4,86
PPL-MCTS - $c_{puct} = 5.0, \tau = 1.0$	0,88	0,55	4,9
PPL-MCTS - $c_{puct} = 5.0, \tau = 1.1$	0,89	0,54	4,97
PPL-MCTS - $c_{puct} = 5.0, \tau = 1.2$	0,89	0,54	4,91
PPL-MCTS - $c_{puct} = 8.0, \tau = 1.0$	0,83	0,54	4,98
PPL-MCTS - $c_{puct} = 8.0, \tau = 1.1$	0,86	0,54	4,95
PPL-MCTS - $c_{puct} = 8.0, \tau = 1.2$	0,88	0,55	4,94

Table A.2 – Results for every tested set of parameters on the proposed methods; CLS dataset. Results reported in the main body are in italic.

Generation method	Accuracy \uparrow	5 - Self-Bleu \downarrow	Oracle perplexity \downarrow
Beam sampling - Argmax $\tau = 1.0$	0,94	0,79	3,55
Beam sampling - Argmax $\tau = 1.1$	0,96	0,77	3,65
<i>Beam sampling - Argmax $\tau = 1.2$</i>	<i>0,97</i>	<i>0,73</i>	<i>3,82</i>
Beam sampling - First true $\tau = 1.0$	0,86	0,77	3,73
Beam sampling - First true $\tau = 1.1$	0,89	0,77	3,68
<i>Beam sampling - First true $\tau = 1.2$</i>	<i>0,9</i>	<i>0,73</i>	<i>3,84</i>
Beam sampling - Sampling $\tau = 1.0$	0,87	0,77	3,7
<i>Beam sampling - Sampling $\tau = 1.1$</i>	<i>0,92</i>	<i>0,76</i>	<i>3,68</i>
Beam sampling - Sampling $\tau = 1.2$	0,89	0,73	3,83
<i>CC-LM - Greedy Search</i>	<i>0,91</i>	<i>0,71</i>	<i>3,21</i>
CC-LM - Sampling $\tau = 1.0$	0,87	0,17	15,7
CC-LM - Sampling $\tau = 1.1$	0,86	0,1	32,2
CC-LM - Sampling $\tau = 1.2$	0,8	0,08	80,2
<i>CC-LM - Classloss - Greedy Search</i>	<i>0,82</i>	<i>0,79</i>	<i>2,56</i>
CC-LM - Classloss - Sampling $\tau = 1.0$	0,81	0,16	18,4
CC-LM - Classloss - Sampling $\tau = 1.1$	0,79	0,1	37,1
CC-LM - Classloss - Sampling $\tau = 1.2$	0,74	0,07	95,4
<i>Sampling - Argmax $\tau = 1.0$</i>	<i>0,99</i>	<i>0,17</i>	<i>16,5</i>
Sampling - Argmax $\tau = 1.1$	0,99	0,11	31,8
Sampling - Argmax $\tau = 1.2$	0,99	0,07	84,50
Sampling - First true $\tau = 1.0$	0,88	0,16	16,4
Sampling - First true $\tau = 1.1$	0,87	0,1	31,5
<i>Sampling - First true $\tau = 1.2$</i>	<i>0,89</i>	<i>0,07</i>	<i>85,9</i>
<i>Sampling - Sampling $\tau = 1.0$</i>	<i>0,88</i>	<i>0,17</i>	<i>16,3</i>
Sampling - Sampling $\tau = 1.1$	0,87	0,1	30,8
Sampling - Sampling $\tau = 1.2$	0,88	0,07	81
PPL-MCTS - $c_{puct} = 1.0, \tau = 1.0$	0,96	0,62	5,61
PPL-MCTS - $c_{puct} = 1.0, \tau = 1.1$	0,96	0,63	5,65
PPL-MCTS - $c_{puct} = 1.0, \tau = 1.2$	0,96	0,62	5,66
<i>PPL-MCTS - $c_{puct} = 3.0, \tau = 1.0$</i>	<i>0,97</i>	<i>0,63</i>	<i>5,69</i>
PPL-MCTS - $c_{puct} = 3.0, \tau = 1.1$	0,97	0,62	5,77
PPL-MCTS - $c_{puct} = 3.0, \tau = 1.2$	0,96	0,62	5,72
PPL-MCTS - $c_{puct} = 5.0, \tau = 1.0$	0,95	0,63	5,6
PPL-MCTS - $c_{puct} = 5.0, \tau = 1.1$	0,96	0,63	5,66
PPL-MCTS - $c_{puct} = 5.0, \tau = 1.2$	0,96	0,63	5,63
PPL-MCTS - $c_{puct} = 8.0, \tau = 1.0$	0,93	0,64	5,57
PPL-MCTS - $c_{puct} = 8.0, \tau = 1.1$	0,93	0,64	5,57
PPL-MCTS - $c_{puct} = 8.0, \tau = 1.2$	0,95	0,63	5,57

Table A.3 – Results for every tested set of parameters on the proposed methods; amazon_polarity dataset. Results reported in the main body are in italic.



GENERATIVE COOPERATIVE NETWORKS FOR NATURAL LANGUAGE GENERATION

B.1 Proof for Theorem 6.2.1

Let p_{data} the data distribution that we seek at approximating, and p_0 be the initial generator, of same support \mathcal{Y} as p_{data} and which is not null everywhere p_{data} is not null.

As considered in Algorithm 1, let consider each step t the learning the following discriminator optimization:

$$D_t \leftarrow \arg \max_{D \in \mathcal{D}} \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log D(y)] + \mathbb{E}_{y \sim p_{t-1}(y)} [\log(1 - D(y))]$$

Thus, following the proof in [121], if \mathcal{D} has enough capacity, $D_t(y) = \frac{p_{\text{data}}(y)}{p_{\text{data}}(y) + p_{t-1}(y)}$ for every $y \in \mathcal{Y}$.

Also, at each step t of Algorithm 1, we set:

$$p_t \leftarrow \arg \min_{p \in \mathcal{G}} KL(q_t || p)$$

With $q_t(y) \triangleq \frac{D_t(y)p_{t-1}(y)}{z_t}$ for each $t > 0$ and all $y \in \mathcal{Y}$, where z_t is the partition function

of distribution q_t .

Thus, if \mathcal{G} has enough capacity and p_t is sufficiently trained, we have for every $y \in \mathcal{Y}$ and every $t \geq 1$:

$$p_t(y) \propto D_t(y)p_{t-1}(y) = \frac{p_{\text{data}}(y)p_{t-1}(y)}{p_{\text{data}}(y) + p_{t-1}(y)} = \frac{p_{\text{data}}(y)}{(p_{\text{data}}(y)/p_{t-1}(y)) + 1} \triangleq \tilde{p}_t(y) \quad (\text{B.1})$$

With $z_t \triangleq \sum_{y \in \mathcal{Y}} \tilde{p}_t(y)$, we have $p_t(y) = \frac{\tilde{p}_t(y)}{z_t}$.

In the following, we consider, for all $y \in \mathcal{Y}$, the sequence $\hat{z}_t(y)$ defined as:

$$\hat{z}_t(y) = \begin{cases} p_{\text{data}}(y)/p_0(y), & \text{if } t = 0; \\ z_t(\hat{z}_{t-1}(y) + 1), & \forall t \geq 1. \end{cases}$$

Lemma B.1.1. *At every step t of Algorithm 1, we have for all $y \in \mathcal{Y}$:*

$$\tilde{p}_{t+1}(y) = \frac{p_{\text{data}}(y)}{\hat{z}_t(y) + 1}$$

Proof. Let consider a proof by induction.

First consider the base case where $t = 0$. From Equation B.1, we have $\tilde{p}_1(y) = \frac{p_{\text{data}}(y)}{(p_{\text{data}}(y)/p_0(y)) + 1}$ and thus, $\tilde{p}_1(y) = \frac{p_{\text{data}}(y)}{\hat{z}_0(y) + 1}$.

Let now assume that $\tilde{p}_t(y) = \frac{p_{\text{data}}(y)}{\hat{z}_{t-1}(y) + 1}$ is true at any step $t > 0$. We need to show that this relation still holds for $t + 1$ to prove the lemma.

Under this assumption, starting from Equation B.1, we have:

$$\begin{aligned} \tilde{p}_{t+1} &= \frac{p_{\text{data}}(y)p_t(y)}{p_{\text{data}}(y) + p_t(y)} = \frac{p_{\text{data}}(y)\tilde{p}_t(y)}{p_{\text{data}}(y)z_t + \tilde{p}_t(y)} = \frac{p_{\text{data}}(y)\tilde{p}_t(y)}{p_{\text{data}}(y)z_t + p_{\text{data}}(y)/(\hat{z}_{t-1}(y) + 1)} \\ &= \frac{\tilde{p}_t(y)(\hat{z}_{t-1}(y) + 1)}{z_t(\hat{z}_{t-1}(y) + 1) + 1} = \frac{p_{\text{data}}(y)}{z_t(\hat{z}_{t-1}(y) + 1) + 1} = \frac{p_{\text{data}}(y)}{\hat{z}_t(y) + 1} \end{aligned}$$

□

Lemma B.1.2. *For every step $t > 1$ of Algorithm 1, $z_t < 1$.*

Proof. For every step $t > 0$, using Lemma B.1.1 on the second and fourth equality (below),

we have:

$$\begin{aligned}
z_{t+1} &= \sum_{y \in \mathcal{Y}} \tilde{p}_{t+1}(y) = \sum_{y \in \mathcal{Y}} \frac{p_{\text{data}}(y)}{\hat{z}_t(y) + 1} = \sum_{y \in \mathcal{Y}} \frac{p_{\text{data}}(y)}{\hat{z}_{t-1}(y) + 1} \frac{\hat{z}_{t-1}(y) + 1}{\hat{z}_t(y) + 1} = \sum_{y \in \mathcal{Y}} \tilde{p}_t(y) \frac{\hat{z}_{t-1}(y) + 1}{\hat{z}_t(y) + 1} \\
&= \sum_{y \in \mathcal{Y}} p_t(y) \frac{z_t(\hat{z}_{t-1}(y) + 1)}{\hat{z}_t(y) + 1} = \sum_{y \in \mathcal{Y}} p_t(y) \frac{\hat{z}_t(y)}{\hat{z}_t(y) + 1} = \mathbb{E}_{y \sim p_t(y)} \left[\frac{\hat{z}_t(y)}{\hat{z}_t(y) + 1} \right]
\end{aligned}$$

Thus, since $\hat{z}_t(y) \geq 0$ for all $y \in \mathcal{Y}$ and all $t \geq 0$, $z_{t+1} < 1$ for all $t > 0$.

□

Then, to prove theorem 1 (convergence of p_t to p_{data} in law), let us rewrite \hat{z}_t (using its definition for $t > 0$) as:

$$\hat{z}_t(y) = z_t(\hat{z}_{t-1}(y) + 1) = \prod_{s=1}^t z_s \left(\frac{p_{\text{data}}(y)}{p_0(y)} \right) + \sum_{s=1}^t \prod_{s'=s}^t z_{s'}$$

For any pair $(y, y') \in \mathcal{Y}^2$, we thus have:

$$\hat{z}_t(y) - \hat{z}_t(y') = \left(\frac{p_{\text{data}}(y)}{p_0(y)} - \frac{p_{\text{data}}(y')}{p_0(y')} \right) \prod_{s=1}^t z_s$$

Since from Lemma B.1.2 we know that $z_t < 1$ for any $t > 1$, we have: $\lim_{t \rightarrow +\infty} \prod_{s=1}^t z_s = 0$ and thus, $\hat{z}_t(y) - \hat{z}_t(y')$ converges to 0 for any pair $(y, y') \in \mathcal{Y}^2$, ensuring that $\hat{z}_t(y)$ converges to a constant K , which shows that

$$\tilde{p}_t(y) \xrightarrow{+\infty} \frac{p_{\text{data}}(y)}{1 + K}$$

which in turn implies our final conclusion, i.e., that p_t converges in distribution to p_{data} .

B.2 Proof for Theorem 6.2.2

Let us consider the case of $p_t \propto p_{t-1} D_t$, and a discriminator sufficiently trained such that, i.e., such that for

$$\log \eta = \min \left(\mathbb{E}_{y \sim p_{\text{data}}(y)} [\log(D_t(y))], \mathbb{E}_{y \sim p_{t-1}(y)} [\log(1 - D_t(y))] \right) \quad (\text{B.2})$$

we have $\eta \in]\frac{1}{2}; 1[$

The difference of KL divergences of the target distribution p_{data} from the generator distribution taken at two successive steps is given as:

$$\begin{aligned}
\Delta_t &\triangleq KL(p_{\text{data}}||p_t) - KL(p_{\text{data}}||p_{t-1}) \\
&= \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log(p_{t-1}(y)) - \log(p_t(y))] \\
&= \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log(p_{t-1}(y)) - \log(p_{t-1}(y)D_t(y))] + \log\left(\sum_{y' \in \mathcal{Y}} p_{t-1}(y)D_t(y)\right) \\
&= \mathbb{E}_{y \sim p_{\text{data}}(y)} [-\log(D_t(y))] + \log\left(\sum_{y \in \mathcal{Y}} p_{t-1}(y)D_t(y)\right) \\
&= \log\left(\mathbb{E}_{y \sim p_{t-1}(y)} [D_t(y)]\right) - \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log(D_t(y))]
\end{aligned}$$

From the assumption given in Equation B.2, we have:

$$\begin{aligned}
\log \eta &\leq \mathbb{E}_{y \sim p_{t-1}(y)} [\log(1 - D_t(y))] \\
&\leq \log\left(\mathbb{E}_{y \sim p_{t-1}(y)} [1 - D_t(y)]\right)
\end{aligned}$$

where the second inequality is obtained with the Jensen inequality on expectations of concave functions.

This equivalent to:

$$\log\left(1 - \mathbb{E}_{y \sim p_{t-1}(y)} [1 - D_t(y)]\right) \leq \log(1 - \eta)$$

And thus:

$$\log\left(\mathbb{E}_{y \sim p_{t-1}(y)} [D_t(y)]\right) \leq \log(1 - \eta)$$

From assumption of Equation B.2, we also know that $\mathbb{E}_{y \sim p_{\text{data}}(y)} [\log(D_t(y))] \geq \log(\eta)$.

Thus, we have:

$$\Delta_t \leq \log(1 - \eta) - \log(\eta) = \log\left(\frac{1}{\eta} - 1\right) < 0$$

which concludes the proof.

B.3 Results with Former MLE Baseline

As mentioned in Section 6.4, results reported in Table 6.1 are not using the same base model as for Figures 6.3 and 6.4. For completeness, we report here results obtained using this former model used for the figures, which has the same architecture but uses a learning rate of $1e-3$ during 2 training epochs, while the new one used for Table 6.1 uses a learning rate of $1e-4$ during 5 training epochs (which is slower but more accurate).

	QG			Summarization		
	B	R-1	R-L	B	R-1	R-L
MLE	16.5	43.9	40	11.5	36.8	34.9
ColdGAN	16.9	44.2	40.3	11.6	37.8	36.4
SelfGAN	17.2	44.3	40.6	12.3	38.6	36.7
GAN $\hat{q}=p$ +scheduler	16.2	43.1	39.3	11.2	36.1	34.3
GAN $\hat{q}=p$	9.4	25.0	22.8	7.1	21.0	19.9
GCN $\hat{q}=p$	16.5	43.9	40.0	11.5	36.8	34.9
GAN $\hat{q}=Nucleus$ +scheduler	16.9	45.0	41.0	11.6	37.7	35.7
GAN $\hat{q}=Nucleus$	9.5	25.3	23.0	7.4	21.2	20.1
GCN $\hat{q}=Nucleus$	17.5	45.3	42.4	12	39.0	37.0
GAN $\hat{q}=MCTS$ +scheduler	17.1	45.5	41.5	11.9	38.1	36.2
GAN $\hat{q}=MCTS$	9.8	26.1	23.8	8.5	21.9	20.7
GCN $\hat{q}=MCTS$	18	45.9	42.6	12.4	39.1	37.1
GCN $\hat{q}=MCTS$ decod=mcts	18.4	46.3	43.1	12.7	39.4	37.4
GCN $\hat{q}=MCTS$ T5-3B	21.8	49.8	45.9	19.2	44.2	43.8

Table B.1 – Final results on QG and Summarization test sets, in terms of BLEU-4 (B), ROUGE-1 (R-1) and ROUGE-L (R-L), using the same base model as used in [305].



THERAPY: GLOBAL EXPLANATION OF TEXTUAL DISCRIMINATIVE MODELS THROUGH COOPERATIVE GENERATION

C.1 Qualitative Results

We provide samples generated by Therapy as well as the first 20 top words returned by the method for the different classes of both datasets. As explained in Section 7.4, available implementations of LIME and SHAP does not allow to study the impact of n-grams. Thus, even if Therapy allows to do it, we restricted the study to unigrams, so some "words" correspond to sub-words, due to the breakdown into unigrams (ve, ll, ...).

amazon_polarity, "positive" class

Samples:

- "The amount of support they gave was amazing. I'm so happy with the game and it's a great way for me to play my favorite role! Also, I love that they're giving us a new character every week, which is awesome :)I think they've been pretty good lately too. It's nice

when we see someone who really loves their team and has fun doing things"

- "The whole point of this post is to show people what a great job you've put in. You are doing an excellent job, I love this sub. Keep going. :) I'm so happy for all those people that have helped! It's amazing to be here and it was fun being the one that got the most attention on the other day :). Thank you so much <3 You're very welcome :) And yes... I am glad you enjoyed this subreddit - it is a great place"

Top words: *great, love, good, ve, years, people, lot, friends, fun, life, world, works, easy, things, happy, heard, including, awesome, nice, family*

amazon_polarity, "negative" class

Samples:

- "I guess you could say it has to be the most boring thing ever. It is a very boring experience I don't think they have any idea how to make money off this, so they will probably just do what you say they will if it doesn't work. They may try to sell you on the fact that it isn't working and then try and get you to buy back into it when they are done making money. If I was them, would they not want to pay me?"
- "It might be a bug, but it doesn't work on my phone Yeah that sucks! I was hoping the same thing happened. I don't think you can get a refund without having to buy a new one... You have no idea what they would be like without the warranty. They wouldn't know how much money is left if they didn't do it properly or at all and they would probably just give it back as soon as they found out it was defective (or something)"

Top words: *don, money, bad, doesn, didn, idea, work, device, isn, thing, guess, wrong, back, buy, fact, time, phone, point, problem, thought*

ag_news, "world" class

Samples:

-
- The world's most popular and sought after luxury home in Dubai, the Al Khaleel International Airport (ALI), is a city that is renowned as the hub of the Middle East and North Africa region. It is also the capital city of Abu Dhabi, United Arab Emirates with a population over 2 billion people and a GDP per person. Al-Khaleel airport is an important gateway into Europe to the north east and west of Europe, Asia, and Australia
 - In the last month, the US government has been accused of using a "secret" military intelligence agency in Iraq and Afghanistan for spying on its allies, including Pakistan, Turkey, Iran, Russia, Syria, Sudan, Lebanon, Libya, Egypt, Nigeria...and even Saudi Arabia! According to a new report published today, the Pentagon is now investigating the alleged use by the US military Intelligence Community (IUC), which was created by President Bush after the 9/11 terror attack that killed over

Top words: *people, man, country, city, party, killed, family, agree, wrong, general, children, sex, president, police, working, military, dead, missing, woman, days*

ag_news, "sport" class

Samples:

- I think it's pretty safe to assume that the guy has a lot of experience with the game. He played in the NHL, and he was a top 10 player on the team for most games last season (he had a goal in his final three playoff series), and he won a Stanley Cup as a rookie this past season (he finished third in the league in points scored, which was good for second in the league) and is still one of the best players in hockey at this stage in the year
- We've got to keep playing this game. This team needs to win games and we need to play the best basketball that's been in our league all season, every night. And it is time to get out there and do that." "The Warriors have won three straight games at home and are 2-0 against the NBA's best team on both sides of the ball this season with a record of 21-1 (13.7 points per game)

Top words: *time, game, back, season, play, didn, team, guy, field, night, games, left,*

12, title, won, saturday, playing, great, day, wasn

ag_news, "business" class

Samples:

- I am still in shock after hearing of that. It's a pretty big deal. It happened last month. They are trying to get the money out of the company by selling their stock for profit so they can sell more shares and buy more shares at higher prices (which I think would have helped with the stock market) and it was reported as an "investment fraud" by the SEC which has been going on all over this subreddit for months, but no one ever seems to care much
- Biden is planning to spend millions of dollars to buy a new home, but the real estate market in America is still struggling with the housing shortage. The average house sale cost \$1 billion and was up by nearly 50 percent from the previous year's price of about \$800 million - according to the Real Estate Board of New York (RBE). The RBE estimates that the average house sales prices are expected to rise 1,000 per month this fiscal year as the economy continues its rebound

Top words: *money, buy, care, doesn, things, deal, pay, worth, business, car, biggest, interested, month, trade, don, compagny, happened, store, kind, price*

ag_news, "sci/tech" class

Samples:

- 2K Games' Dark Souls 3 is coming to PC, Mac & Linux in the near future. The new game will launch for free on PC, Mac & Linux and Xbox One, PlayStation 5 and Microsoft Windows, as well. It'll come out sometime during this week, with an official release expected soon thereafter, though we don't yet know what it will be called or where exactly you're getting the title. We also have some news from Sony that's not quite so surprising etc...
- In this new age of technology, the world needs more people. We have a lot in our hands. The internet can help us connect to others

through video chat and online games.""The company will launch a mobile game called 'Gangster', where it plans to offer ""an interactive experience"" with its users, according to the company. The game has been developed for the Apple iPad and Android phones that use Apple TV, which also uses Google Chromecast, according to a release.

Top words: *ve, ll, idea, phone, internet, make, system, video, online, life, understand, version, pc, found, 13, thing, computer, lot, hard, issue, people, work, information, future*

BIBLIOGRAPHY

- [1] Scott Aaronson and Hendrik Kirchner, *Watermarking GPT Outputs*, 2023, URL: <https://www.scottaaronson.com/talks/watermark.ppt>.
- [2] Sahar Abdelnabi, Rakibul Hasan, and Mario Fritz, « Open-Domain, Content-based, Multi-modal Fact-checking of Out-of-Context Images via Online Resources », in: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, IEEE, 2022, pp. 14920–14929, DOI: [10.1109/CVPR52688.2022.01452](https://doi.org/10.1109/CVPR52688.2022.01452), URL: <https://doi.org/10.1109/CVPR52688.2022.01452>.
- [3] Abubakar Abid, Maheen Farooqi, and James Zou, « Persistent Anti-Muslim Bias in Large Language Models », in: *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, ACM, 2021, pp. 298–306, URL: <https://doi.org/10.1145/3461702.3462624>.
- [4] Shruti Agarwal et al., « Protecting World Leaders Against Deep Fakes », in: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, 2019, pp. 38–45, URL: http://openaccess.thecvf.com/content/_CVPRW/_2019/html/Media/_Forensics/Agarwal/_Protecting/_World/_Leaders/_Against/_Deep/_Fakes/_CVPRW/_2019/_paper.html.
- [5] Stability AI, *Meet Stable Beluga 1 and Stable Beluga 2, Our Large and Mighty Instruction Fine-Tuned Language Models*, <https://stability.ai/blog/stable-beluga-large-instruction-fine-tuned-models>, Accessed: 2023-07-31, 2023.
- [6] Firoj Alam et al., « Fighting the COVID-19 Infodemic in Social Media: A Holistic Perspective and a Call to Arms », in: *Proceedings of the Fifteenth International AAAI Conference on Web and Social Media, ICWSM 2021, held virtually, June 7-10, 2021*, ed. by Ceren Budak et al., AAAI Press, 2021, pp. 913–922, URL: <https://ojs.aaai.org/index.php/ICWSM/article/view/18114>.

-
- [7] Hunt Allcott and Matthew Gentzkow, « Social Media and Fake News in the 2016 Election », in: *Journal of Economic Perspectives* 31.2 (May 2017), pp. 211–36, DOI: [10.1257/jep.31.2.211](https://doi.org/10.1257/jep.31.2.211), URL: <https://www.aeaweb.org/articles?id=10.1257/jep.31.2.211>.
- [8] Adam L. Alter and Daniel M. Oppenheimer, « Uniting the Tribes of Fluency to Form a Metacognitive Nation », in: *Personality and Social Psychology Review* 13.3 (2009), PMID: 19638628, pp. 219–235, DOI: [10.1177/1088868309341564](https://doi.org/10.1177/1088868309341564), eprint: <https://doi.org/10.1177/1088868309341564>, URL: <https://doi.org/10.1177/1088868309341564>.
- [9] Bárbara G. Amado, Ramón Arce, and Francisca Fariña, « Undeutsch hypothesis and Criteria Based Content Analysis: A meta-analytic review », in: *The European Journal of Psychology Applied to Legal Context* 7.1 (2015), pp. 3–12, ISSN: 1889-1861, DOI: <https://doi.org/10.1016/j.ejpal.2014.11.002>, URL: <https://www.sciencedirect.com/science/article/pii/S1889186114000183>.
- [10] Ali Amin-Nejad, Julia Ive, and Sumithra Velupillai, « Exploring Transformer Text Generation for Medical Dataset Augmentation », English, in: *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France: European Language Resources Association, May 2020, pp. 4699–4708, ISBN: 979-10-95546-34-4, URL: <https://www.aclweb.org/anthology/2020.lrec-1.578>.
- [11] Peter Anderson et al., « SPICE: Semantic Propositional Image Caption Evaluation », in: *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, ed. by Bastian Leibe et al., vol. 9909, Lecture Notes in Computer Science, Springer, 2016, pp. 382–398, DOI: [10.1007/978-3-319-46454-1_24](https://doi.org/10.1007/978-3-319-46454-1_24), URL: https://doi.org/10.1007/978-3-319-46454-1_24.
- [12] Galen Andrew et al., « Deep Canonical Correlation Analysis », in: *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, vol. 28, JMLR Workshop and Conference Proceedings, JMLR.org, 2013, pp. 1247–1255, URL: <http://proceedings.mlr.press/v28/andrew13.html>.
- [13] Shivangi Aneja, Christoph Bregler, and Matthias Nießner, « Catching Out-of-Context Misinformation with Self-supervised Learning », in: *CoRR* abs/2101.06278 (2021), arXiv: [2101.06278](https://arxiv.org/abs/2101.06278), URL: <https://arxiv.org/abs/2101.06278>.

-
- [14] Thomas Anthony, Zheng Tian, and David Barber, « Thinking fast and slow with deep learning and tree search », in: *arXiv preprint arXiv:1705.08439* (2017).
- [15] Wissam Antoun, Benoît Sagot, and Djamé Seddah, *From Text to Source: Results in Detecting Large Language Model-Generated Content*, 2023, arXiv: [2309.13322 \[cs.CL\]](#).
- [16] Wissam Antoun et al., « Towards a Robust Detection of Language Model-Generated Text: Is ChatGPT that easy to detect? », in: *30e Conférence sur le Traitement Automatique des Langues Naturelles, ATALA*, 2023, pp. 14–27.
- [17] Michał Araszkiewicz et al., « Thirty years of Artificial Intelligence and Law: overviews », in: *Artificial Intelligence and Law* (Aug. 2022), ISSN: 1572-8382, URL: <https://doi.org/10.1007/s10506-022-09324-9>.
- [18] Martín Arjovsky and Léon Bottou, « Towards Principled Methods for Training Generative Adversarial Networks », in: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, Open-Review.net, 2017, URL: https://openreview.net/forum?id=Hk4_qw5xe.
- [19] Alejandro Barredo Arrieta et al., « Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI », in: *Inf. Fusion* 58 (2020), pp. 82–115, URL: <https://doi.org/10.1016/j.inffus.2019.12.012>.
- [20] Natalia Aruguete et al., « Truth be told: How “true” and “false” labels influence user engagement with fact-checks », in: *New Media and Society* (Sept. 2023), DOI: [10.1177/14614448231193709](https://doi.org/10.1177/14614448231193709).
- [21] Michael Bommarito II and Daniel Martin Katz, *GPT Takes the Bar Exam*, 2022, arXiv: [2212.14402 \[cs.CL\]](#).
- [22] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton, « Layer Normalization », in: *CoRR* abs/1607.06450 (2016), arXiv: [1607.06450](#), URL: <http://arxiv.org/abs/1607.06450>.
- [23] Sameer Badaskar, Sachin Agarwal, and Shilpa Arora, « Identifying Real or Fake Articles: Towards better Language Modeling », in: *Third International Joint Conference on Natural Language Processing, IJCNLP 2008, Hyderabad, India, January 7-12, 2008*, The Association for Computer Linguistics, 2008, pp. 817–822, URL: <https://aclanthology.org/I08-2115/>.

-
- [24] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, « Neural Machine Translation by Jointly Learning to Align and Translate », in: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, ed. by Yoshua Bengio and Yann LeCun, 2015, URL: <http://arxiv.org/abs/1409.0473>.
- [25] Anton Bakhtin et al., « Real or Fake? Learning to Discriminate Machine from Human Generated Text », in: *CoRR* abs/1906.03351 (2019), arXiv: [1906.03351](https://arxiv.org/abs/1906.03351), URL: <http://arxiv.org/abs/1906.03351>.
- [26] Ramy Baly et al., « Predicting Factuality of Reporting and Bias of News Media Sources », in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, ed. by Ellen Riloff et al., Association for Computational Linguistics, 2018, pp. 3528–3539, DOI: [10.18653/v1/d18-1389](https://doi.org/10.18653/v1/d18-1389), URL: <https://doi.org/10.18653/v1/d18-1389>.
- [27] Satanjeev Banerjee and Alon Lavie, « METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments », in: *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, ed. by Jade Goldstein et al., Association for Computational Linguistics, 2005, pp. 65–72, URL: <https://aclanthology.org/W05-0909/>.
- [28] Hangbo Bao et al., « BEiT: BERT Pre-Training of Image Transformers », in: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022, URL: <https://openreview.net/forum?id=p-BhZSz59o4>.
- [29] Hangbo Bao et al., « UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training », in: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, vol. 119, Proceedings of Machine Learning Research, PMLR, 2020, pp. 642–652, URL: <http://proceedings.mlr.press/v119/bao20a.html>.
- [30] Hongyan Bao et al., « Towards Understanding the Robustness Against Evasion Attack on Categorical Data », in: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022, URL: <https://openreview.net/forum?id=BmJV7kyAmg>.

-
- [31] Sudipta Basu, « The conservatism principle and the asymmetric timeliness of earnings¹ », in: *Journal of Accounting and Economics* 24.1 (1997), Properties of Accounting Earnings, pp. 3–37, ISSN: 0165-4101, DOI: [https://doi.org/10.1016/S0165-4101\(97\)00014-1](https://doi.org/10.1016/S0165-4101(97)00014-1), URL: <https://www.sciencedirect.com/science/article/pii/S0165410197000141>.
- [32] Belhassen Bayar and Matthew C. Stamm, « A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer », in: *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, IH&MMSec 2016, Vigo, Galicia, Spain, June 20-22, 2016*, ed. by Fernando Pérez-González et al., ACM, 2016, pp. 5–10, DOI: [10.1145/2909827.2930786](https://doi.org/10.1145/2909827.2930786), URL: <https://doi.org/10.1145/2909827.2930786>.
- [33] Samy Bengio et al., « Scheduled sampling for sequence prediction with recurrent neural networks », in: *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.
- [34] Yoshua Bengio et al., « A Neural Probabilistic Language Model », in: *J. Mach. Learn. Res.* 3 (2003), pp. 1137–1155, URL: <http://jmlr.org/papers/v3/bengio03a.html>.
- [35] James Betker et al., « Improving Image Generation with Better Captions », in: *OpenAI Blog* (2023).
- [36] Tiziano Bianchi and Alessandro Piva, « Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts », in: *IEEE Trans. Inf. Forensics Secur.* 7.3 (2012), pp. 1003–1017, DOI: [10.1109/TIFS.2012.2187516](https://doi.org/10.1109/TIFS.2012.2187516), URL: <https://doi.org/10.1109/TIFS.2012.2187516>.
- [37] Amirhosein Bodaghi and Jonice Oliveira, « The characteristics of rumor spreaders on Twitter: A quantitative analysis on real data », in: *Comput. Commun.* 160 (2020), pp. 674–687, DOI: [10.1016/j.comcom.2020.07.017](https://doi.org/10.1016/j.comcom.2020.07.017), URL: <https://doi.org/10.1016/j.comcom.2020.07.017>.
- [38] Francesco Bodria et al., « Benchmarking and Survey of Explanation Methods for Black Box Models », in: *CoRR* (2021), URL: <https://arxiv.org/abs/2102.13076>.
- [39] Lawrence E. Boehm, « The Validity Effect: A Search for Mediating Variables », in: *Personality and Social Psychology Bulletin* 20.3 (1994), pp. 285–293, DOI: [10.1177/1461621694203001](https://doi.org/10.1177/1461621694203001).

-
- 0146167294203006, eprint: <https://doi.org/10.1177/0146167294203006>, URL: <https://doi.org/10.1177/0146167294203006>.
- [40] Christina Boididou et al., « Challenges of computational verification in social multi-media », in: *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume*, ed. by Chin-Wan Chung et al., ACM, 2014, pp. 743–748, DOI: [10.1145/2567948.2579323](https://doi.org/10.1145/2567948.2579323), URL: <https://doi.org/10.1145/2567948.2579323>.
- [41] Christina Boididou et al., « Verifying Multimedia Use at MediaEval 2016 », in: *Working Notes Proceedings of the MediaEval 2016 Workshop, Hilversum, The Netherlands, October 20-21, 2016*, ed. by Guillaume Gravier et al., vol. 1739, CEUR Workshop Proceedings, CEUR-WS.org, 2016, URL: https://ceur-ws.org/Vol-1739/MediaEval_2016_paper_3.pdf.
- [42] Sebastian Borgeaud et al., « Improving Language Models by Retrieving from Trillions of Tokens », in: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ed. by Kamalika Chaudhuri et al., vol. 162, Proceedings of Machine Learning Research, PMLR, 2022, pp. 2206–2240, URL: <https://proceedings.mlr.press/v162/borgeaud22a.html>.
- [43] Alexandre Bovet and Hernán A. Makse, « Influence of fake news in Twitter during the 2016 US presidential election », in: *CoRR abs/1803.08491 (2018)*, arXiv: [1803.08491](https://arxiv.org/abs/1803.08491), URL: <http://arxiv.org/abs/1803.08491>.
- [44] Lia Bozarth and Ceren Budak, « Toward a Better Performance Evaluation Framework for Fake News Classification », in: *Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, ICWSM 2020, Held Virtually, Original Venue: Atlanta, Georgia, USA, June 8-11, 2020*, ed. by Munmun De Choudhury et al., AAAI Press, 2020, pp. 60–71, URL: <https://ojs.aaai.org/index.php/ICWSM/article/view/7279>.
- [45] Steven Bramhall et al., « QLIME-A: Quadratic Local Interpretable Model-Agnostic Explanation Approach », in: *SMU Data Science Rev 3 (2020)*.
- [46] Tom B. Brown et al., « Language Models are Few-Shot Learners », in: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, ed. by Hugo Larochelle et al., 2020, URL: <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html>.

-
- [47] Varun H Buch, Irfan Ahmed, and Mahiben Maruthappu, « Artificial intelligence in medicine: current trends and future possibilities », en, in: *Br. J. Gen. Pract.* 68.668 (Mar. 2018), pp. 143–144.
- [48] Emanuele Bugliarello et al., « Multimodal Pretraining Unmasked: A Meta-Analysis and a Unified Framework of Vision-and-Language BERTs », in: *Trans. Assoc. Comput. Linguistics* 9 (2021), pp. 978–994, DOI: [10.1162/tacl_a_00408](https://doi.org/10.1162/tacl_a_00408), URL: https://doi.org/10.1162/tacl_a_00408.
- [49] Massimo Caccia et al., « Language GANs Falling Short », in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020, URL: <https://openreview.net/forum?id=BJgza6VtPB>.
- [50] Rémi Cadène et al., « RUBi: Reducing Unimodal Biases for Visual Question Answering », in: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, ed. by Hanna M. Wallach et al., 2019, pp. 839–850, URL: <https://proceedings.neurips.cc/paper/2019/hash/51d92be1c60d1db1d2e5e7a07da55b26-Abstract.html>.
- [51] Chiyu Cai, Linjing Li, and Daniel Zeng, « Detecting Social Bots by Jointly Modeling Deep Behavior and Content Information », in: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, ed. by Ee-Peng Lim et al., ACM, 2017, pp. 1995–1998, DOI: [10.1145/3132847.3133050](https://doi.org/10.1145/3132847.3133050), URL: <https://doi.org/10.1145/3132847.3133050>.
- [52] Juan Cao et al., « Exploring the Role of Visual Content in Fake News Detection », in: *CoRR* abs/2003.05096 (2020), arXiv: [2003.05096](https://arxiv.org/abs/2003.05096), URL: <https://arxiv.org/abs/2003.05096>.
- [53] Nicholas Carlini et al., « Extracting Training Data from Large Language Models », in: *arXiv* (2020), arXiv: [2012.07805](https://arxiv.org/abs/2012.07805) [cs.CR].
- [54] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso, « Machine Learning Interpretability: A Survey on Methods and Metrics », in: *Electronics* 8.8 (2019), ISSN: 2079-9292, DOI: [10.3390/electronics8080832](https://doi.org/10.3390/electronics8080832), URL: <https://www.mdpi.com/2079-9292/8/8/832>.

-
- [55] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete, « Information credibility on twitter », in: *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, ed. by Sadagopan Srinivasan et al., ACM, 2011, pp. 675–684, DOI: [10.1145/1963405.1963500](https://doi.org/10.1145/1963405.1963500), URL: <https://doi.org/10.1145/1963405.1963500>.
- [56] Souradip Chakraborty et al., « On the Possibilities of AI-Generated Text Detection », in: *CoRR abs/2304.04736* (2023), DOI: [10.48550/arXiv.2304.04736](https://doi.org/10.48550/arXiv.2304.04736), arXiv: [2304.04736](https://doi.org/10.48550/arXiv.2304.04736), URL: <https://doi.org/10.48550/arXiv.2304.04736>.
- [57] Soravit Changpinyo et al., « Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts », in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, Computer Vision Foundation / IEEE, 2021, pp. 3558–3568, DOI: [10.1109/CVPR46437.2021.00356](https://openaccess.thecvf.com/content/CVPR2021/html/Changpinyo_Conceptual_12M_Pushing_Web-Scale_Image-Text_Pre-Training_To_Recognize_Long-Tail_Visual_CVPR_2021_paper.html), URL: https://openaccess.thecvf.com/content/CVPR2021/html/Changpinyo_Conceptual_12M_Pushing_Web-Scale_Image-Text_Pre-Training_To_Recognize_Long-Tail_Visual_CVPR_2021_paper.html.
- [58] Tong Che et al., « Maximum-Likelihood Augmented Discrete Generative Adversarial Networks », in: *CoRR abs/1702.07983* (2017), arXiv: [1702.07983](http://arxiv.org/abs/1702.07983), URL: <http://arxiv.org/abs/1702.07983>.
- [59] Gullal S. Cheema, Sherzod Hakimov, and Ralph Ewerth, « TIB's Visual Analytics Group at MediaEval '20: Detecting Fake News on Corona Virus and 5G Conspiracy », in: *MediaEval Benchmarking Initiative for Multimedia Evaluation (MediaEval 2020)*, online, United States, Dec. 2020.
- [60] Beidi Chen et al., « Scatterbrain: Unifying Sparse and Low-rank Attention », in: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, ed. by Marc'Aurelio Ranzato et al., 2021, pp. 17413–17426, URL: <https://proceedings.neurips.cc/paper/2021/hash/9185f3ec501c674c7c788464a36e7fb3-Abstract.html>.
- [61] Xingyuan Chen et al., « Adding A Filter Based on The Discriminator to Improve Unconditional Text Generation », in: *arXiv preprint arXiv:2004.02135* (2020).

-
- [62] Xinru Chen et al., « Image Manipulation Detection by Multi-View Multi-Scale Supervision », in: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, IEEE, 2021, pp. 14165–14173, DOI: [10.1109/ICCV48922.2021.01392](https://doi.org/10.1109/ICCV48922.2021.01392), URL: <https://doi.org/10.1109/ICCV48922.2021.01392>.
- [63] Yen-Chun Chen et al., « UNITER: UNiversal Image-TExt Representation Learning », in: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXX*, ed. by Andrea Vedaldi et al., vol. 12375, Lecture Notes in Computer Science, Springer, 2020, pp. 104–120, DOI: [10.1007/978-3-030-58577-8_7](https://doi.org/10.1007/978-3-030-58577-8_7), URL: https://doi.org/10.1007/978-3-030-58577-8_7.
- [64] Yixuan Chen et al., « Cross-modal Ambiguity Learning for Multimodal Fake News Detection », in: *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, ed. by Frédérique Laforest et al., ACM, 2022, pp. 2897–2905, DOI: [10.1145/3485447.3511968](https://doi.org/10.1145/3485447.3511968), URL: <https://doi.org/10.1145/3485447.3511968>.
- [65] Jaemin Cho et al., « Fine-grained Image Captioning with CLIP Reward », in: *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, Association for Computational Linguistics, 2022, pp. 517–527, DOI: [10.18653/v1/2022.findings-naacl.39](https://doi.org/10.18653/v1/2022.findings-naacl.39), URL: <https://doi.org/10.18653/v1/2022.findings-naacl.39>.
- [66] Krzysztof Marcin Choromanski et al., « Rethinking Attention with Performers », in: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021, URL: <https://openreview.net/forum?id=Ua6zuk0WRH>.
- [67] Leshem Choshen et al., « On the Weaknesses of Reinforcement Learning for Neural Machine Translation », in: *International Conference on Learning Representations*, 2020, URL: <https://openreview.net/forum?id=H1eCw3EKvH>.
- [68] Miranda Christ, Sam Gunn, and Or Zamir, « Undetectable Watermarks for Language Models », in: *IACR Cryptol. ePrint Arch. (2023)*, p. 763, URL: <https://eprint.iacr.org/2023/763>.

-
- [69] Giovanni Luca Ciampaglia et al., « Computational fact checking from knowledge networks », in: *CoRR* abs/1501.03471 (2015), arXiv: [1501.03471](https://arxiv.org/abs/1501.03471), URL: <http://arxiv.org/abs/1501.03471>.
- [70] Matteo Cinelli et al., « Echo Chambers on Social Media: A comparative analysis », in: *CoRR* abs/2004.09603 (2020), arXiv: [2004.09603](https://arxiv.org/abs/2004.09603), URL: <https://arxiv.org/abs/2004.09603>.
- [71] Elizabeth Clark et al., « All That's 'Human' Is Not Gold: Evaluating Human Evaluation of Generated Text », in: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, ed. by Chengqing Zong et al., Association for Computational Linguistics, 2021, pp. 7282–7296, DOI: [10.18653/v1/2021.acl-long.565](https://doi.org/10.18653/v1/2021.acl-long.565), URL: <https://doi.org/10.18653/v1/2021.acl-long.565>.
- [72] Jack Clark, Alec Radford, and Jeff Wu, *GPT-2 detection baselines*, <https://github.com/openai/gpt-2-output-dataset/blob/master/detection.md>, Accessed: 2023-07-13, 2019.
- [73] Vincent Claveau, « Detecting fake news in tweets from text and propagation graph: IRISA's participation to the FakeNews task at MediaEval 2020 », in: *MediaEval Benchmarking Initiative for Multimedia Evaluation (MediaEval 2020)*, online, United States, Dec. 2020, URL: <https://hal.archives-ouvertes.fr/hal-03116027>.
- [74] Ronan Collobert et al., « Natural Language Processing (Almost) from Scratch », in: *J. Mach. Learn. Res.* 12 (2011), pp. 2493–2537, DOI: [10.5555/1953048.2078186](https://doi.org/10.5555/1953048.2078186), URL: <https://dl.acm.org/doi/10.5555/1953048.2078186>.
- [75] Content European Commission. Directorate-General for Communication Networks and Technology, *A Multi-dimensional Approach to Disinformation: Report of the Independent High Level Group on Fake News and Online Disinformation*, Publications Office of the European Union, 2018, ISBN: 9789279804205, URL: <https://books.google.fr/books?id=i8uQtQEACAAJ>.
- [76] Alba Cotarelo et al., « Improving Monte Carlo Tree Search with Artificial Neural Networks without Heuristics », in: *Applied Sciences* 11 (Feb. 2021), p. 2056, DOI: [10.3390/app11052056](https://doi.org/10.3390/app11052056).

-
- [77] Rémi Coulom, « Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search », in: *Computers and Games, 5th International Conference, CG 2006, Turin, Italy, May 29-31, 2006. Revised Papers*, ed. by H. Jaap van den Herik, Paolo Ciancarini, and H. H. L. M. Donkers, vol. 4630, Lecture Notes in Computer Science, Springer, 2006, pp. 72–83, DOI: [10.1007/978-3-540-75538-8_7](https://doi.org/10.1007/978-3-540-75538-8_7), URL: https://doi.org/10.1007/978-3-540-75538-8_7.
- [78] Evan Crothers et al., « Adversarial Robustness of Neural-Statistical Features in Detection of Generative Transformers », in: *International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022*, IEEE, 2022, pp. 1–8, DOI: [10.1109/IJCNN55064.2022.9892269](https://doi.org/10.1109/IJCNN55064.2022.9892269), URL: <https://doi.org/10.1109/IJCNN55064.2022.9892269>.
- [79] Bo Dai and Dahua Lin, « Contrastive Learning for Image Captioning », in: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, ed. by Isabelle Guyon et al., 2017, pp. 898–907, URL: <https://proceedings.neurips.cc/paper/2017/hash/46922a0880a8f11f8f69cbb52b1396be-Abstract.html>.
- [80] Bo Dai et al., « Towards Diverse and Natural Image Descriptions via a Conditional GAN », in: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, IEEE Computer Society, 2017, pp. 2989–2998, DOI: [10.1109/ICCV.2017.323](https://doi.org/10.1109/ICCV.2017.323), URL: <https://doi.org/10.1109/ICCV.2017.323>.
- [81] Mayur Datar et al., « Locality-sensitive hashing scheme based on p-stable distributions », in: *Proceedings of the 20th ACM Symposium on Computational Geometry, Brooklyn, New York, USA, June 8-11, 2004*, ed. by Jack Snoeyink and Jean-Daniel Boissonnat, ACM, 2004, pp. 253–262, DOI: [10.1145/997817.997857](https://doi.org/10.1145/997817.997857), URL: <https://doi.org/10.1145/997817.997857>.
- [82] Sumanth Dathathri et al., « Plug and Play Language Models: A Simple Approach to Controlled Text Generation », in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020, URL: <https://openreview.net/forum?id=H1edEyBKDS>.
- [83] Jonas De keersmaecker and Arne Roets, « ‘Fake news’: Incorrect, but hard to correct. The role of cognitive ability on the impact of false information on social impressions », in: *Intelligence* 65 (2017), pp. 107–110, ISSN: 0160-2896, DOI: <https://doi.org/>

-
- 10.1016/j.intell.2017.10.005, URL: <https://www.sciencedirect.com/science/article/pii/S0160289617301617>.
- [84] Yuntian Deng et al., « Residual Energy-Based Models for Text Generation », in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020, URL: <https://openreview.net/forum?id=B1l4SgHKDH>.
- [85] Carnegie-Mellon University.Computer Science Dept., *Speech understanding systems: summary of results of the five-year research effort at Carnegie-Mellon University*. June 2018, DOI: 10.1184/R1/6609821.v1, URL: https://kilthub.cmu.edu/articles/journal_contribution/Speech_understanding_systems_summary_of_results_of_the_five-year_research_effort_at_Carnegie-Mellon_University_/6609821/1.
- [86] Tim Dettmers and Luke Zettlemoyer, *The case for 4-bit precision: k-bit Inference Scaling Laws*, 2023, arXiv: 2212.09720 [cs.LG].
- [87] Tim Dettmers et al., *LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale*, 2022, arXiv: 2208.07339 [cs.LG].
- [88] Jacob Devlin et al., « BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding », in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, ed. by Jill Burstein, Christy Doran, and Thamar Solorio, Association for Computational Linguistics, 2019, pp. 4171–4186, DOI: 10.18653/v1/n19-1423, URL: <https://doi.org/10.18653/v1/n19-1423>.
- [89] Shimin Di, Yanyan Shen, and Lei Chen, « Relation Extraction via Domain-aware Transfer Learning », in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, ed. by Ankur Teredesai et al., ACM, 2019, pp. 1348–1357, DOI: 10.1145/3292500.3330890, URL: <https://doi.org/10.1145/3292500.3330890>.
- [90] Li Dong et al., « Unified Language Model Pre-training for Natural Language Understanding and Generation », in: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, ed. by Hanna M. Wallach et al., 2019,

-
- pp. 13042–13054, URL: <https://proceedings.neurips.cc/paper/2019/hash/c20bb2d9a50d5ac1f713f8b34d9aac5a-Abstract.html>.
- [91] Alexey Dosovitskiy et al., « An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale », in: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021, URL: <https://openreview.net/forum?id=YicbFdNTTy>.
- [92] Yao Dou et al., « Is GPT-3 Text Indistinguishable from Human Text? Scarecrow: A Framework for Scrutinizing Machine Text », in: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, Association for Computational Linguistics, 2022, pp. 7250–7274, DOI: [10.18653/v1/2022.acl-long.501](https://doi.org/10.18653/v1/2022.acl-long.501), URL: <https://doi.org/10.18653/v1/2022.acl-long.501>.
- [93] Matthijs Douze et al., « The 2021 Image Similarity Dataset and Challenge », in: *CoRR abs/2106.09672* (2021), arXiv: [2106.09672](https://arxiv.org/abs/2106.09672), URL: <https://arxiv.org/abs/2106.09672>.
- [94] Tyna Eloundou et al., *GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models*, 2023, arXiv: [2303.10130](https://arxiv.org/abs/2303.10130) [econ.GN].
- [95] Radwa ElShawi et al., « ILIME: Local and Global Interpretable Model-Agnostic Explainer of Black-Box Decision », in: *ADBS*, 2019.
- [96] Dumitru Erhan et al., « Why Does Unsupervised Pre-training Help Deep Learning? », in: *J. Mach. Learn. Res.* 11 (2010), pp. 625–660, DOI: [10.5555/1756006.1756025](https://doi.org/10.5555/1756006.1756025), URL: <https://dl.acm.org/doi/10.5555/1756006.1756025>.
- [97] Patrick Esser et al., *Structure and Content-Guided Video Synthesis with Diffusion Models*, 2023, arXiv: [2302.03011](https://arxiv.org/abs/2302.03011) [cs.CV].
- [98] Andre Esteva et al., « Dermatologist-level classification of skin cancer with deep neural networks », in: *nature* 542.7639 (2017), pp. 115–118.
- [99] Alexander Fabbri et al., « Improving Zero and Few-Shot Abstractive Summarization with Intermediate Fine-tuning and Data Augmentation », in: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Lin-

-
- guistics, June 2021, pp. 704–717, DOI: [10.18653/v1/2021.naacl-main.57](https://doi.org/10.18653/v1/2021.naacl-main.57), URL: <https://aclanthology.org/2021.naacl-main.57>.
- [100] Angela Fan, Mike Lewis, and Yann N. Dauphin, « Hierarchical Neural Story Generation », in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, ed. by Iryna Gurevych and Yusuke Miyao, Association for Computational Linguistics, 2018, pp. 889–898, DOI: [10.18653/v1/P18-1082](https://doi.org/10.18653/v1/P18-1082), URL: <https://aclanthology.org/P18-1082/>.
- [101] Song Feng, Ritwik Banerjee, and Yejin Choi, « Syntactic Stylometry for Deception Detection », in: *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, The Association for Computer Linguistics, 2012, pp. 171–175, URL: <https://aclanthology.org/P12-2034/>.
- [102] Elise Fenn et al., « Nonprobative Photos Increase Truth, Like, and Share Judgments in a Simulated Social Media Environment », in: *Journal of Applied Research in Memory and Cognition* 8.2 (2019), pp. 131–138, ISSN: 2211-3681, DOI: <https://doi.org/10.1016/j.jarmac.2019.04.005>, URL: <https://www.sciencedirect.com/science/article/pii/S2211368118302936>.
- [103] Elise Fenn et al., « The effect of nonprobative photographs on truthiness persists over time », in: *Acta Psychologica* 144.1 (2013), pp. 207–211, ISSN: 0001-6918, DOI: <https://doi.org/10.1016/j.actpsy.2013.06.004>, URL: <https://www.sciencedirect.com/science/article/pii/S0001691813001376>.
- [104] Pierre Fernandez et al., « Three Bricks to Consolidate Watermarks for Large Language Models », in: *IEEE International Workshop on Information Forensics and Security, WIFS 2023, Nuremberg, Germany, December 4-7, 2023*, IEEE, 2023.
- [105] Pierre Fernandez et al., « Watermarking Images in Self-Supervised Latent Spaces », in: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022.
- [106] Emilio Ferrara, « Disinformation and social bot operations in the run up to the 2017 French presidential election », in: *First Monday* 22.8 (2017), DOI: [10.5210/fm.v22i8.8005](https://doi.org/10.5210/fm.v22i8.8005), URL: <https://doi.org/10.5210/fm.v22i8.8005>.

-
- [107] Emilio Ferrara et al., « The rise of social bots », in: *Commun. ACM* 59.7 (2016), pp. 96–104, DOI: [10.1145/2818717](https://doi.org/10.1145/2818717), URL: <https://doi.org/10.1145/2818717>.
- [108] Robert J. Fisher, « Social Desirability Bias and the Validity of Indirect Questioning », in: *Journal of Consumer Research* 20.2 (Sept. 1993), pp. 303–315, ISSN: 0093-5301, DOI: [10.1086/209351](https://doi.org/10.1086/209351), eprint: <https://academic.oup.com/jcr/article-pdf/20/2/303/5074014/20-2-303.pdf>, URL: <https://doi.org/10.1086/209351>.
- [109] Joel Frank et al., « Leveraging Frequency Analysis for Deep Fake Image Recognition », in: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, vol. 119, Proceedings of Machine Learning Research, PMLR, 2020, pp. 3247–3258, URL: <http://proceedings.mlr.press/v119/frank20a.html>.
- [110] Stella Frank, Emanuele Bugliarello, and Desmond Elliott, « Vision-and-Language or Vision-for-Language? On Cross-Modal Influence in Multimodal Transformers », in: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, ed. by Marie-Francine Moens et al., Association for Computational Linguistics, 2021, pp. 9847–9857, DOI: [10.18653/v1/2021.emnlp-main.775](https://doi.org/10.18653/v1/2021.emnlp-main.775), URL: <https://doi.org/10.18653/v1/2021.emnlp-main.775>.
- [111] Leon Fröhling and Arkaitz Zubiaga, « Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover », in: *PeerJ Comput. Sci.* 7 (2021), e443, DOI: [10.7717/peerj-cs.443](https://doi.org/10.7717/peerj-cs.443), URL: <https://doi.org/10.7717/peerj-cs.443>.
- [112] Philip Gage, « A new algorithm for data compression », in: *The C Users Journal archive* 12 (1994), pp. 23–38.
- [113] Siddhant Garg, Thuy Vu, and Alessandro Moschitti, « TANDA: Transfer and Adapt Pre-Trained Transformer Models for Answer Sentence Selection », in: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, AAAI Press, 2020, pp. 7780–7788, URL: <https://ojs.aaai.org/index.php/AAAI/article/view/6282>.

-
- [114] Romaric Gaudel et al., « s-LIME: Reconciling Locality and Fidelity in Linear Explanations », in: *Advances in Intelligent Data Analysis XX - 20th International Symposium on Intelligent Data Analysis, IDA 2022, Rennes, France, April 20-22, 2022, Proceedings*, vol. 13205, Lecture Notes in Computer Science, Springer, 2022, pp. 102–114, URL: https://doi.org/10.1007/978-3-031-01333-1_9.
- [115] Weifeng Ge et al., « Deep Metric Learning with Hierarchical Triplet Loss », in: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI*, ed. by Vittorio Ferrari et al., vol. 11210, Lecture Notes in Computer Science, Springer, 2018, pp. 272–288, DOI: [10.1007/978-3-030-01231-1_17](https://doi.org/10.1007/978-3-030-01231-1_17), URL: https://doi.org/10.1007/978-3-030-01231-1_17.
- [116] Jonas Gehring et al., « Convolutional Sequence to Sequence Learning », in: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ed. by Doina Precup and Yee Whye Teh, vol. 70, Proceedings of Machine Learning Research, PMLR, 2017, pp. 1243–1252, URL: <http://proceedings.mlr.press/v70/gehring17a.html>.
- [117] Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush, « GLTR: Statistical Detection and Visualization of Generated Text », in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Florence, Italy: Association for Computational Linguistics, July 2019, pp. 111–116, DOI: [10.18653/v1/P19-3019](https://www.aclweb.org/anthology/P19-3019), URL: <https://www.aclweb.org/anthology/P19-3019>.
- [118] Faeze Ghorbanpour et al., *FNR: A Similarity and Transformer-Based Approach to Detect Multi-Modal Fake News in Social Media*, 2021, arXiv: [2112.01131](https://arxiv.org/abs/2112.01131) [cs.MM].
- [119] Josh A. Goldstein et al., *Generative Language Models and Automated Influence Operations: Emerging Threats and Potential Mitigations*, 2023, arXiv: [2301.04246](https://arxiv.org/abs/2301.04246) [cs.CY].
- [120] Miroslav Goljan, Jessica J. Fridrich, and Mo Chen, « Defending Against Fingerprint-Copy Attack in Sensor-Based Camera Identification », in: *IEEE Trans. Inf. Forensics Secur.* 6.1 (2011), pp. 227–236, DOI: [10.1109/TIFS.2010.2099220](https://doi.org/10.1109/TIFS.2010.2099220), URL: <https://doi.org/10.1109/TIFS.2010.2099220>.
- [121] Ian J. Goodfellow et al., « Generative Adversarial Nets », in: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, ed. by Zoubin Ghahramani et al., 2014, pp. 2672–2680, URL: <https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html>.

-
- [122] Shreya Goyal et al., *A Survey of Adversarial Defences and Robustness in NLP*, 2023, arXiv: 2203.06414 [cs.CL].
- [123] Yash Goyal et al., « Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering », in: *Int. J. Comput. Vis.* 127.4 (2019), pp. 398–414, DOI: 10.1007/s11263-018-1116-0, URL: <https://doi.org/10.1007/s11263-018-1116-0>.
- [124] Ciara Greene and Gillian Murphy, « Quantifying the effects of fake news on behavior: Evidence from a study of COVID-19 misinformation », in: *Journal of experimental psychology: Applied*, ACM, 2019, 773—784, DOI: 10.1037/xap0000371, URL: <https://doi.org/10.1037/xap0000371>.
- [125] Riccardo Guidotti, « Evaluating local explanation methods on ground truth », in: *Artif. Intell.* 291 (2021), p. 103428, URL: <https://doi.org/10.1016/j.artint.2020.103428>.
- [126] Qipeng Guo et al., « Star-Transformer », in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, ed. by Jill Burstein, Christy Doran, and Tamar Solorio, Association for Computational Linguistics, 2019, pp. 1315–1325, DOI: 10.18653/v1/n19-1133, URL: <https://doi.org/10.18653/v1/n19-1133>.
- [127] Wenzhong Guo, Jianwen Wang, and Shiping Wang, « Deep Multimodal Representation Learning: A Survey », in: *IEEE Access* 7 (2019), pp. 63373–63394, DOI: 10.1109/ACCESS.2019.2916887, URL: <https://doi.org/10.1109/ACCESS.2019.2916887>.
- [128] Zhijiang Guo, Michael Sejr Schlichtkrull, and Andreas Vlachos, « A Survey on Automated Fact-Checking », in: *Trans. Assoc. Comput. Linguistics* 10 (2022), pp. 178–206, DOI: 10.1162/tacl_a_00454, URL: https://doi.org/10.1162/tacl_a_00454.
- [129] Aditi Gupta et al., « Faking Sandy: characterizing and identifying fake images on Twitter during Hurricane Sandy », in: *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*, ed. by Leslie Carr et al., International World Wide Web Conferences Steering Committee / ACM, 2013, pp. 729–736, DOI: 10.1145/2487788.2488033, URL: <https://doi.org/10.1145/2487788.2488033>.

-
- [130] Kelvin Guu et al., « REALM: Retrieval-Augmented Language Model Pre-Training », in: *Proceedings of the 37th International Conference on Machine Learning, ICML'20*, JMLR.org, 2020.
- [131] Michael Hameleers et al., « A Picture Paints a Thousand Lies? The Effects and Mechanisms of Multimodal Disinformation and Rebuttals Disseminated via Social Media », in: *Political Communication* 37.2 (2020), pp. 281–301, DOI: [10.1080/10584609.2019.1674979](https://doi.org/10.1080/10584609.2019.1674979), eprint: <https://doi.org/10.1080/10584609.2019.1674979>, URL: <https://doi.org/10.1080/10584609.2019.1674979>.
- [132] Ben Harwood et al., « Smart Mining for Deep Metric Learning », in: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, IEEE Computer Society, 2017, pp. 2840–2848, DOI: [10.1109/ICCV.2017.307](https://doi.org/10.1109/ICCV.2017.307), URL: <https://doi.org/10.1109/ICCV.2017.307>.
- [133] Di He et al., « Decoding with value networks for neural machine translation », in: *Advances in Neural Information Processing Systems* 30 (2017).
- [134] Kaiming He et al., « Deep Residual Learning for Image Recognition », in: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, pp. 770–778, DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90), URL: <https://doi.org/10.1109/CVPR.2016.90>.
- [135] Pengcheng He, Jianfeng Gao, and Weizhu Chen, « DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing », in: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, OpenReview.net, 2023, URL: <https://openreview.net/pdf?id=sE7-XhLxHA>.
- [136] Silvan Heller, Luca Rossetto, and Heiko Schuldt, « The PS-Battles Dataset - an Image Collection for Image Manipulation Detection », in: *CoRR* abs/1804.04866 (2018), arXiv: [1804.04866](https://arxiv.org/abs/1804.04866), URL: <http://arxiv.org/abs/1804.04866>.
- [137] Lisa Anne Hendricks et al., « Decoupling the Role of Data, Attention, and Losses in Multimodal Transformers », in: *Trans. Assoc. Comput. Linguistics* 9 (2021), pp. 570–585, DOI: [10.1162/tac1_a_00385](https://doi.org/10.1162/tac1_a_00385), URL: https://doi.org/10.1162/tac1_a_00385.

-
- [138] Linda Henkel, « Photograph-induced memory errors: When photographs make people claim they have done things they have not », in: *Applied Cognitive Psychology* 25 (Jan. 2011), pp. 78–86, DOI: [10.1002/acp.1644](https://doi.org/10.1002/acp.1644).
- [139] G. E. Hinton and R. R. Salakhutdinov, « Reducing the Dimensionality of Data with Neural Networks », in: *Science* 313.5786 (2006), pp. 504–507, DOI: [10.1126/science.1127647](https://doi.org/10.1126/science.1127647), eprint: <https://www.science.org/doi/pdf/10.1126/science.1127647>, URL: <https://www.science.org/doi/abs/10.1126/science.1127647>.
- [140] Geoffrey E. Hinton, James L. McClelland, and David E. Rumelhart, « Distributed Representations », in: *The Philosophy of Artificial Intelligence*, ed. by Margaret A. Boden, Oxford readings in philosophy, Oxford University Press, 1990, pp. 248–280.
- [141] Jonathan Ho et al., *Imagen Video: High Definition Video Generation with Diffusion Models*, 2022, arXiv: [2210.02303](https://arxiv.org/abs/2210.02303) [cs.CV].
- [142] Sepp Hochreiter and Jürgen Schmidhuber, « Long Short-Term Memory », in: *Neural Computation* 9.8 (1997), pp. 1735–1780, DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735), URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [143] Ari Holtzman et al., « Learning to Write with Cooperative Discriminators », in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, ed. by Iryna Gurevych and Yusuke Miyao, Association for Computational Linguistics, 2018, pp. 1638–1649, DOI: [10.18653/v1/P18-1152](https://doi.org/10.18653/v1/P18-1152), URL: <https://www.aclweb.org/anthology/P18-1152/>.
- [144] Ari Holtzman et al., « The Curious Case of Neural Text Degeneration », in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020, URL: <https://openreview.net/forum?id=rygGQyrFvH>.
- [145] Ukyo Honda, Taro Watanabe, and Yuji Matsumoto, « Switching to Discriminative Image Captioning by Relieving a Bottleneck of Reinforcement Learning », in: *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, IEEE, 2023, pp. 1124–1134, DOI: [10.1109/WACV56688.2023.00118](https://doi.org/10.1109/WACV56688.2023.00118), URL: <https://doi.org/10.1109/WACV56688.2023.00118>.

-
- [146] Or Honovich et al., « Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor », in: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 14409–14428, DOI: [10.18653/v1/2023.acl-long.806](https://doi.org/10.18653/v1/2023.acl-long.806), URL: <https://aclanthology.org/2023.acl-long.806>.
- [147] Edward J. Hu et al., *LoRA: Low-Rank Adaptation of Large Language Models*, 2021, arXiv: [2106.09685](https://arxiv.org/abs/2106.09685) [cs.CL].
- [148] Linmei Hu et al., « Deep learning for fake news detection: A comprehensive survey », in: *AI Open 3* (2022), pp. 133–155, DOI: [10.1016/j.aiopen.2022.09.001](https://doi.org/10.1016/j.aiopen.2022.09.001), URL: <https://doi.org/10.1016/j.aiopen.2022.09.001>.
- [149] Xiaowei Hu et al., « Scaling Up Vision-Language Pre-Training for Image Captioning », in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 17980–17989.
- [150] Drew A. Hudson and Christopher D. Manning, « GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering », in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, 2019, pp. 6700–6709, DOI: [10.1109/CVPR.2019.00686](https://doi.org/10.1109/CVPR.2019.00686), URL: http://openaccess.thecvf.com/content/_CVPR/_2019/html/Hudson_GQA_A_New_Dataset_for_Real-World_Visual_Reasoning_and_Compositional_CVPR_2019_paper.html.
- [151] Daphne Ippolito et al., « Automatic Detection of Generated Text is Easiest when Humans are Fooled », in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, ed. by Dan Jurafsky et al., Association for Computational Linguistics, 2020, pp. 1808–1822, DOI: [10.18653/v1/2020.acl-main.164](https://doi.org/10.18653/v1/2020.acl-main.164), URL: <https://doi.org/10.18653/v1/2020.acl-main.164>.
- [152] Cherilyn Ireton and Julie Posetti, « Journalism, fake news & disinformation : handbook for journalism education and training / », in: (2018), Includes bibliographical references., 128 p. : URL: <http://digitallibrary.un.org/record/1641987>.
- [153] Md Saiful Islam et al., « COVID-19 vaccine rumors and conspiracy theories: The need for cognitive inoculation against misinformation to improve vaccine adherence », in: *PLOS ONE 16.5* (May 2021), pp. 1–17, DOI: [10.1371/journal.pone.0251605](https://doi.org/10.1371/journal.pone.0251605), URL: <https://doi.org/10.1371/journal.pone.0251605>.

-
- [154] Gautier Izacard et al., « Few-shot Learning with Retrieval Augmented Language Models », in: *CoRR* abs/2208.03299 (2022), DOI: [10.48550/arXiv.2208.03299](https://doi.org/10.48550/arXiv.2208.03299), arXiv: [2208.03299](https://doi.org/10.48550/arXiv.2208.03299), URL: <https://doi.org/10.48550/arXiv.2208.03299>.
- [155] Alon Jacovi, « Trends in Explainable AI (XAI) Literature », in: *CoRR* abs/2301.05433 (2023), DOI: [10.48550/arXiv.2301.05433](https://doi.org/10.48550/arXiv.2301.05433), arXiv: [2301.05433](https://doi.org/10.48550/arXiv.2301.05433), URL: <https://doi.org/10.48550/arXiv.2301.05433>.
- [156] Ayush Jaiswal et al., « AIRD: Adversarial Learning Framework for Image Repurposing Detection », in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, 2019, pp. 11330–11339, DOI: [10.1109/CVPR.2019.01159](https://doi.org/10.1109/CVPR.2019.01159), URL: http://openaccess.thecvf.com/content/_CVPR/_2019/html/Jaiswal/_AIRD_Adversarial_Learning_Framework_for_Image_Repurposing_Detection_CVPR_2019_paper.html.
- [157] Ayush Jaiswal et al., « Multimedia Semantic Integrity Assessment Using Joint Embedding Of Images And Text », in: *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*, ed. by Qiong Liu et al., ACM, 2017, pp. 1465–1471, DOI: [10.1145/3123266.3123385](https://doi.org/10.1145/3123266.3123385), URL: <https://doi.org/10.1145/3123266.3123385>.
- [158] Eric Jang, Shixiang Gu, and Ben Poole, « Categorical Reparameterization with Gumbel-Softmax », in: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017, URL: <https://openreview.net/forum?id=rkE3y85ee>.
- [159] Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan, « Automatic Detection of Machine Generated Text: A Critical Survey », in: *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, ed. by Donia Scott, Núria Bel, and Chengqing Zong, International Committee on Computational Linguistics, 2020, pp. 2296–2309, DOI: [10.18653/v1/2020.coling-main.208](https://doi.org/10.18653/v1/2020.coling-main.208), URL: <https://doi.org/10.18653/v1/2020.coling-main.208>.
- [160] Hervé Jégou, Matthijs Douze, and Cordelia Schmid, « Product Quantization for Nearest Neighbor Search », in: *IEEE Trans. Pattern Anal. Mach. Intell.* 33.1 (2011), pp. 117–128, DOI: [10.1109/TPAMI.2010.57](https://doi.org/10.1109/TPAMI.2010.57), URL: <https://doi.org/10.1109/TPAMI.2010.57>.

-
- [161] Haoming Jiang et al., « SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization », in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, July 2020, pp. 2177–2190, DOI: [10.18653/v1/2020.acl-main.197](https://doi.org/10.18653/v1/2020.acl-main.197), URL: <https://www.aclweb.org/anthology/2020.acl-main.197>.
- [162] Xiaoqi Jiao et al., « TinyBERT: Distilling BERT for Natural Language Understanding », in: *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, Nov. 2020, pp. 4163–4174, DOI: [10.18653/v1/2020.findings-emnlp.372](https://doi.org/10.18653/v1/2020.findings-emnlp.372), URL: <https://aclanthology.org/2020.findings-emnlp.372>.
- [163] Zhiwei Jin et al., « Multimodal Fusion with Recurrent Neural Networks for Rumor Detection on Microblogs », in: *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*, ed. by Qiong Liu et al., ACM, 2017, pp. 795–816, DOI: [10.1145/3123266.3123454](https://doi.org/10.1145/3123266.3123454), URL: <https://doi.org/10.1145/3123266.3123454>.
- [164] Zhiwei Jin et al., « Novel Visual and Statistical Image Features for Microblogs News Verification », in: *IEEE Transactions on Multimedia* 19.3 (2017), pp. 598–608, DOI: [10.1109/TMM.2016.2617078](https://doi.org/10.1109/TMM.2016.2617078).
- [165] Sarthak Jindal et al., « NewsBag: A Benchmark Multimodal Dataset for Fake News Detection », in: *Proceedings of the Workshop on Artificial Intelligence Safety, co-located with 34th AAAI Conference on Artificial Intelligence, SafeAI@AAAI 2020, New York City, NY, USA, February 7, 2020*, ed. by Huáscar Espinoza et al., vol. 2560, CEUR Workshop Proceedings, CEUR-WS.org, 2020, pp. 138–145, URL: <https://ceur-ws.org/Vol-2560/paper27.pdf>.
- [166] Michał Jungiewicz and Aleksander Smywinski-Pohl, « Towards textual data augmentation for neural networks: synonyms and maximum loss », in: *Computer Science* 20.1 (2019), ISSN: 2300-7036, DOI: [10.7494/csci.2019.20.1.3023](https://doi.org/10.7494/csci.2019.20.1.3023), URL: <https://journals.agh.edu.pl/csci/article/view/3023>.
- [167] Jared Kaplan et al., « Scaling Laws for Neural Language Models », in: *CoRR* abs/2001.08361 (2020), arXiv: [2001.08361](https://arxiv.org/abs/2001.08361), URL: <https://arxiv.org/abs/2001.08361>.

-
- [168] P. Karatza et al., « Interpretability methods of machine learning algorithms with applications in breast cancer diagnosis », in: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2021, pp. 2310–2313, DOI: [10.1109/EMBC46164.2021.9630556](https://doi.org/10.1109/EMBC46164.2021.9630556).
- [169] Andrej Karpathy and Li Fei-Fei, « Deep Visual-Semantic Alignments for Generating Image Descriptions », in: *IEEE Trans. Pattern Anal. Mach. Intell.* 39.4 (2017), pp. 664–676, DOI: [10.1109/TPAMI.2016.2598339](https://doi.org/10.1109/TPAMI.2016.2598339), URL: <https://doi.org/10.1109/TPAMI.2016.2598339>.
- [170] Angelos Katharopoulos et al., « Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention », in: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, vol. 119, Proceedings of Machine Learning Research, PMLR, 2020, pp. 5156–5165, URL: <http://proceedings.mlr.press/v119/katharopoulos20a.html>.
- [171] Sahar Kazemzadeh et al., « ReferItGame: Referring to Objects in Photographs of Natural Scenes », in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans, ACL, 2014, pp. 787–798, DOI: [10.3115/v1/d14-1086](https://doi.org/10.3115/v1/d14-1086), URL: <https://doi.org/10.3115/v1/d14-1086>.
- [172] Nitish Shirish Keskar et al., « CTRL: A Conditional Transformer Language Model for Controllable Generation », in: *CoRR* abs/1909.05858 (2019), arXiv: [1909.05858](https://arxiv.org/abs/1909.05858), URL: <http://arxiv.org/abs/1909.05858>.
- [173] Dhruv Khattar et al., « MVAE: Multimodal Variational Autoencoder for Fake News Detection », in: *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, ed. by Ling Liu et al., ACM, 2019, pp. 2915–2921, DOI: [10.1145/3308558.3313552](https://doi.org/10.1145/3308558.3313552), URL: <https://doi.org/10.1145/3308558.3313552>.
- [174] Prannay Khosla et al., « Supervised Contrastive Learning », in: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, ed. by Hugo Larochelle et al., 2020, URL: <https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html>.

-
- [175] John Kirchenbauer et al., « A Watermark for Large Language Models », in: *CoRR* abs/2301.10226 (2023), DOI: [10.48550/arXiv.2301.10226](https://doi.org/10.48550/arXiv.2301.10226), arXiv: [2301.10226](https://arxiv.org/abs/2301.10226), URL: <https://doi.org/10.48550/arXiv.2301.10226>.
- [176] John Kirchenbauer et al., « On the Reliability of Watermarks for Large Language Models », in: *CoRR* abs/2306.04634 (2023), DOI: [10.48550/arXiv.2306.04634](https://doi.org/10.48550/arXiv.2306.04634), arXiv: [2306.04634](https://arxiv.org/abs/2306.04634), URL: <https://doi.org/10.48550/arXiv.2306.04634>.
- [177] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya, « Reformer: The Efficient Transformer », in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020, URL: <https://openreview.net/forum?id=rkgNKkHtvB>.
- [178] Sosuke Kobayashi, « Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations », in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 452–457, DOI: [10.18653/v1/N18-2072](https://doi.org/10.18653/v1/N18-2072), URL: <https://www.aclweb.org/anthology/N18-2072>.
- [179] Satwik Kottur et al., « Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog », in: *Conference on Empirical Methods in Natural Language Processing*, 2017, URL: <https://api.semanticscholar.org/CorpusID:6683636>.
- [180] Ben Krause et al., « GeDi: Generative Discriminator Guided Sequence Generation », in: *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, ed. by Marie-Francine Moens et al., Association for Computational Linguistics, 2021, pp. 4929–4952, URL: <https://aclanthology.org/2021.findings-emnlp.424>.
- [181] Ranjay Krishna et al., « Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations », in: *Int. J. Comput. Vis.* 123.1 (2017), pp. 32–73, DOI: [10.1007/s11263-016-0981-7](https://doi.org/10.1007/s11263-016-0981-7), URL: <https://doi.org/10.1007/s11263-016-0981-7>.
- [182] Nir Kshetri and Jeffrey M. Voas, « The Economics of "Fake News" », in: *IT Prof.* 19.6 (2017), pp. 8–12, DOI: [10.1109/MITP.2017.4241459](https://doi.org/10.1109/MITP.2017.4241459), URL: <https://doi.org/10.1109/MITP.2017.4241459>.

-
- [183] Rohith Kuditipudi et al., *Robust Distortion-free Watermarks for Language Models*, 2023, arXiv: [2307.15593](https://arxiv.org/abs/2307.15593) [cs.LG].
- [184] Taku Kudo and John Richardson, « SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing », in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, ed. by Eduardo Blanco and Wei Lu, Association for Computational Linguistics, 2018, pp. 66–71, DOI: [10.18653/v1/d18-2012](https://doi.org/10.18653/v1/d18-2012), URL: <https://doi.org/10.18653/v1/d18-2012>.
- [185] Varun Kumar, Ashutosh Choudhary, and Eunah Cho, « Data Augmentation using Pre-trained Transformer Models », in: *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 18–26, URL: <https://www.aclweb.org/anthology/2020.lifelongnlp-1.3>.
- [186] Matt J. Kusner and José Miguel Hernández-Lobato, « GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution », in: *CoRR* abs/1611.04051 (2016), arXiv: [1611.04051](https://arxiv.org/abs/1611.04051), URL: <http://arxiv.org/abs/1611.04051>.
- [187] Sejeong Kwon et al., « Prominent Features of Rumor Propagation in Online Social Media », in: *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, ed. by Hui Xiong et al., IEEE Computer Society, 2013, pp. 1103–1108, DOI: [10.1109/ICDM.2013.61](https://doi.org/10.1109/ICDM.2013.61), URL: <https://doi.org/10.1109/ICDM.2013.61>.
- [188] David M. J. Lazer et al., « The science of fake news », in: *Science* 359.6380 (2018), pp. 1094–1096, DOI: [10.1126/science.aao2998](https://doi.org/10.1126/science.aao2998), eprint: <https://www.science.org/doi/pdf/10.1126/science.aao2998>, URL: <https://www.science.org/doi/abs/10.1126/science.aao2998>.
- [189] Hang Le et al., « FlauBERT: Unsupervised Language Model Pre-training for French », in: *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, ed. by Nicoletta Calzolari et al., European Language Resources Association, 2020, pp. 2479–2490, URL: <https://aclanthology.org/2020.lrec-1.302/>.

-
- [190] Rémi Leblond et al., « Machine Translation Decoding beyond Beam Search », in: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, ed. by Marie-Francine Moens et al., Association for Computational Linguistics, 2021, pp. 8410–8434, DOI: [10.18653/v1/2021.emnlp-main.662](https://doi.org/10.18653/v1/2021.emnlp-main.662), URL: <https://doi.org/10.18653/v1/2021.emnlp-main.662>.
- [191] Yann LeCun et al., « Backpropagation Applied to Handwritten Zip Code Recognition », in: *Neural Comput.* 1.4 (1989), pp. 541–551, DOI: [10.1162/neco.1989.1.4.541](https://doi.org/10.1162/neco.1989.1.4.541), URL: <https://doi.org/10.1162/neco.1989.1.4.541>.
- [192] Sang-Hoon Lee et al., *HierVST: Hierarchical Adaptive Zero-shot Voice Style Transfer*, 2023, arXiv: [2307.16171](https://arxiv.org/abs/2307.16171) [cs.SD].
- [193] H. Leibenstein, « Bandwagon, Snob, and Veblen Effects in the Theory of Consumers' Demand », in: *The Quarterly Journal of Economics* 64.2 (1950), pp. 183–207, ISSN: 00335533, 15314650, URL: <http://www.jstor.org/stable/1882692> (visited on 07/25/2023).
- [194] Mike Lewis et al., « BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension », in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, ed. by Dan Jurafsky et al., Association for Computational Linguistics, 2020, pp. 7871–7880, DOI: [10.18653/v1/2020.acl-main.703](https://doi.org/10.18653/v1/2020.acl-main.703), URL: <https://doi.org/10.18653/v1/2020.acl-main.703>.
- [195] Patrick Lewis et al., « Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks », in: *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Vancouver, BC, Canada: Curran Associates Inc., 2020*, ISBN: 9781713829546.
- [196] Junnan Li et al., « BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation », in: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ed. by Kamalika Chaudhuri et al., vol. 162, Proceedings of Machine Learning Research, PMLR, 2022, pp. 12888–12900, URL: <https://proceedings.mlr.press/v162/li22n.html>.
- [197] Liunian Harold Li et al., « VisualBERT: A Simple and Performant Baseline for Vision and Language », in: *CoRR* abs/1908.03557 (2019), arXiv: [1908.03557](https://arxiv.org/abs/1908.03557), URL: <http://arxiv.org/abs/1908.03557>.

-
- [198] Xiujun Li et al., « Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks », in: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXX*, ed. by Andrea Vedaldi et al., vol. 12375, Lecture Notes in Computer Science, Springer, 2020, pp. 121–137, DOI: [10.1007/978-3-030-58577-8_8](https://doi.org/10.1007/978-3-030-58577-8_8), URL: https://doi.org/10.1007/978-3-030-58577-8_8.
- [199] Yiyi Li and Ying Xie, « Is a Picture Worth a Thousand Words? An Empirical Study of Image Content and Social Media Engagement », in: *Journal of Marketing Research* 57 (2019), pp. 1 –19.
- [200] Yuezun Li, Ming-Ching Chang, and Siwei Lyu, « In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking », in: *CoRR* abs/1806.02877 (2018), arXiv: [1806.02877](http://arxiv.org/abs/1806.02877), URL: <http://arxiv.org/abs/1806.02877>.
- [201] Yuezun Li and Siwei Lyu, « Exposing DeepFake Videos By Detecting Face Warping Artifacts », in: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, 2019, pp. 46–52, URL: http://openaccess.thecvf.com/content/_CVPRW/_2019/html/Media/_Forensics/Li/_Exposing/_DeepFake/_Videos/_By/_Detecting/_Face/_Warping/_Artifacts/_CVPRW/_2019/_paper.html.
- [202] Zhongyang Li, Xiao Ding, and Ting Liu, « Story Ending Prediction by Transferable BERT », in: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, ed. by Sarit Kraus, ijcai.org, 2019, pp. 1800–1806, DOI: [10.24963/ijcai.2019/249](https://doi.org/10.24963/ijcai.2019/249), URL: <https://doi.org/10.24963/ijcai.2019/249>.
- [203] Weixin Liang et al., « GPT detectors are biased against non-native English writers », in: *CoRR* abs/2304.02819 (2023), DOI: [10.48550/arXiv.2304.02819](https://doi.org/10.48550/arXiv.2304.02819), arXiv: [2304.02819](https://doi.org/10.48550/arXiv.2304.02819), URL: <https://doi.org/10.48550/arXiv.2304.02819>.
- [204] Chin-Yew Lin, « Rouge: A package for automatic evaluation of summaries », in: *Text summarization branches out*, 2004, pp. 74–81.
- [205] Hongtao Lin et al., « Learning Dual Retrieval Module for Semi-supervised Relation Extraction », in: *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, ed. by Ling Liu et al., ACM, 2019, pp. 1073–1083, DOI: [10.1145/3308558.3313573](https://doi.org/10.1145/3308558.3313573), URL: <https://doi.org/10.1145/3308558.3313573>.

-
- [206] Tsung-Yi Lin et al., « Microsoft COCO: Common Objects in Context », in: *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, ed. by David J. Fleet et al., vol. 8693, Lecture Notes in Computer Science, Springer, 2014, pp. 740–755, DOI: [10.1007/978-3-319-10602-1_48](https://doi.org/10.1007/978-3-319-10602-1_48), URL: https://doi.org/10.1007/978-3-319-10602-1_48.
- [207] Jiacheng Liu et al., *Making PPO even better: Value-Guided Monte-Carlo Tree Search decoding*, 2023, arXiv: [2309.15028](https://arxiv.org/abs/2309.15028) [cs.CL].
- [208] Peter J. Liu et al., « Generating Wikipedia by Summarizing Long Sequences », in: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018, URL: <https://openreview.net/forum?id=Hyg0vbWC->.
- [209] Yinhan Liu et al., « RoBERTa: A Robustly Optimized BERT Pretraining Approach », in: *CoRR abs/1907.11692* (2019), arXiv: [1907.11692](https://arxiv.org/abs/1907.11692), URL: <http://arxiv.org/abs/1907.11692>.
- [210] Ilya Loshchilov and Frank Hutter, « Decoupled Weight Decay Regularization », in: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019, URL: <https://openreview.net/forum?id=Bkg6RiCqY7>.
- [211] Jiasen Lu et al., « ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks », in: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, ed. by Hanna M. Wallach et al., 2019, pp. 13–23, URL: <https://proceedings.neurips.cc/paper/2019/hash/c74d97b01eae257e44aa9d5bade97baf-Abstract.html>.
- [212] Ximing Lu et al., « NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics », in: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, Association for Computational Linguistics, 2022, pp. 780–799, DOI: [10.18653/v1/2022.naacl-main.57](https://doi.org/10.18653/v1/2022.naacl-main.57), URL: <https://doi.org/10.18653/v1/2022.naacl-main.57>.

-
- [213] Li Lucy and David Bamman, « Gender and Representation Bias in GPT-3 Generated Stories », in: *Proceedings of the Third Workshop on Narrative Understanding*, Virtual: Association for Computational Linguistics, June 2021, pp. 48–55, DOI: [10.18653/v1/2021.nuse-1.5](https://doi.org/10.18653/v1/2021.nuse-1.5), URL: <https://aclanthology.org/2021.nuse-1.5>.
- [214] Scott M. Lundberg and Su-In Lee, « A Unified Approach to Interpreting Model Predictions », in: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, ed. by Isabelle Guyon et al., 2017, pp. 4765–4774, URL: <https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html>.
- [215] Grace Luo, Trevor Darrell, and Anna Rohrbach, « NewsCLIPpings: Automatic Generation of Out-of-Context Multimodal Media », in: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, ed. by Marie-Francine Moens et al., Association for Computational Linguistics, 2021, pp. 6801–6817, DOI: [10.18653/v1/2021.emnlp-main.545](https://doi.org/10.18653/v1/2021.emnlp-main.545), URL: <https://doi.org/10.18653/v1/2021.emnlp-main.545>.
- [216] Ruotian Luo et al., « Discriminability Objective for Training Descriptive Captions », in: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 6964–6974, DOI: [10.1109/CVPR.2018.00728](https://doi.org/10.1109/CVPR.2018.00728), URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Luo/Discriminability_Objective_for_CVPR_2018_paper.html.
- [217] Nishtha Madaan et al., « Generate Your Counterfactuals: Towards Controlled Counterfactual Generation for Text », in: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, AAAI Press, 2021, pp. 13516–13524, URL: <https://ojs.aaai.org/index.php/AAAI/article/view/17594>.
- [218] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh, « The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables », in: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference*

Track Proceedings, OpenReview.net, 2017, URL: <https://openreview.net/forum?id=S1jE5L5gl>.

- [219] Chris J. Maddison, Daniel Tarlow, and Tom Minka, « A* Sampling », in: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, ed. by Zoubin Ghahramani et al., 2014, pp. 3086–3094, URL: <https://proceedings.neurips.cc/paper/2014/hash/309fee4e541e51de2e41f21bebb342aa-Abstract.html>.
- [220] Babak Mahdian and Stanislav Saic, « Using noise inconsistencies for blind image forensics », in: *Image Vis. Comput.* 27.10 (2009), pp. 1497–1503, DOI: [10.1016/j.imavis.2009.02.001](https://doi.org/10.1016/j.imavis.2009.02.001), URL: <https://doi.org/10.1016/j.imavis.2009.02.001>.
- [221] R. Manmatha et al., « Sampling Matters in Deep Embedding Learning », in: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, IEEE Computer Society, 2017, pp. 2859–2867, DOI: [10.1109/ICCV.2017.309](https://doi.org/10.1109/ICCV.2017.309), URL: <https://doi.org/10.1109/ICCV.2017.309>.
- [222] Regina Marchi, « With Facebook, Blogs, and Fake News, Teens Reject Journalistic “Objectivity” », in: *Journal of Communication Inquiry* 36.3 (2012), pp. 246–262, DOI: [10.1177/0196859912458700](https://doi.org/10.1177/0196859912458700), eprint: <https://doi.org/10.1177/0196859912458700>, URL: <https://doi.org/10.1177/0196859912458700>.
- [223] Giovanni Da San Martino et al., « Fine-Grained Analysis of Propaganda in News Article », in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, ed. by Kentaro Inui et al., Association for Computational Linguistics, 2019, pp. 5635–5645, DOI: [10.18653/v1/D19-1565](https://doi.org/10.18653/v1/D19-1565), URL: <https://doi.org/10.18653/v1/D19-1565>.
- [224] Cyprien de Masson d'Autume et al., « Training Language GANs from Scratch », in: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, ed. by Hanna M. Wallach et al., 2019, pp. 4302–4313, URL: <https://proceedings.neurips.cc/paper/2019/hash/a6ea8471c120fe8cc35a2954c9b9c595-Abstract.html>.

-
- [225] Scott McCloskey and Michael Albright, « Detecting GAN-generated Imagery using Color Cues », in: *CoRR* abs/1812.08247 (2018), arXiv: [1812.08247](https://arxiv.org/abs/1812.08247), URL: <http://arxiv.org/abs/1812.08247>.
- [226] Aditya Krishna Menon et al., « In defense of dual-encoders for neural ranking », in: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ed. by Kamalika Chaudhuri et al., vol. 162, Proceedings of Machine Learning Research, PMLR, 2022, pp. 15376–15400, URL: <https://proceedings.mlr.press/v162/menon22a.html>.
- [227] Antoine Miech et al., « Thinking Fast and Slow: Efficient Text-to-Visual Retrieval With Transformers », in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, Computer Vision Foundation / IEEE, 2021, pp. 9826–9836, DOI: [10.1109/CVPR46437.2021.00970](https://doi.org/10.1109/CVPR46437.2021.00970), URL: https://openaccess.thecvf.com/content/CVPR2021/html/Miech_Thinking_Fast_and_Slow_Efficient_Text-to-Visual_Retrieval_With_Transformers_CVPR_2021_paper.html.
- [228] Tomás Mikolov et al., « Distributed Representations of Words and Phrases and their Compositionality », in: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, ed. by Christopher J. C. Burges et al., 2013, pp. 3111–3119, URL: <https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html>.
- [229] Tomás Mikolov et al., « Efficient Estimation of Word Representations in Vector Space », in: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, ed. by Yoshua Bengio and Yann LeCun, 2013, URL: <http://arxiv.org/abs/1301.3781>.
- [230] George A. Miller, « WordNet: A Lexical Database for English », in: *Commun. ACM* 38.11 (Nov. 1995), 39–41, ISSN: 0001-0782, DOI: [10.1145/219717.219748](https://doi.org/10.1145/219717.219748), URL: <https://doi.org/10.1145/219717.219748>.
- [231] Tim Miller, « Explanation in artificial intelligence: Insights from the social sciences », in: *Artif. Intell.* 267 (2019), pp. 1–38, DOI: [10.1016/j.artint.2018.07.007](https://doi.org/10.1016/j.artint.2018.07.007), URL: <https://doi.org/10.1016/j.artint.2018.07.007>.

-
- [232] Yisroel Mirsky and Wenke Lee, « The Creation and Detection of Deepfakes: A Survey », *in: ACM Comput. Surv.* 54.1 (2022), 7:1–7:41, DOI: [10.1145/3425780](https://doi.org/10.1145/3425780), URL: <https://doi.org/10.1145/3425780>.
- [233] Ken Mishima and Hayato Yamana, « A survey on explainable fake news detection », *in: IEICE TRANSACTIONS on Information and Systems* 105.7 (2022), pp. 1249–1257.
- [234] Volodymyr Mnih et al., « Asynchronous Methods for Deep Reinforcement Learning », *in: Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, ed. by Maria-Florina Balcan and Kilian Q. Weinberger, vol. 48, JMLR Workshop and Conference Proceedings, JMLR.org, 2016, pp. 1928–1937, URL: <http://proceedings.mlr.press/v48/mnih16.html>.
- [235] Fred Morstatter et al., « A new approach to bot detection: Striking the balance between precision and recall », *in: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016, San Francisco, CA, USA, August 18-21, 2016*, ed. by Ravi Kumar, James Caverlee, and Hanghang Tong, IEEE Computer Society, 2016, pp. 533–540, DOI: [10.1109/ASONAM.2016.7752287](https://doi.org/10.1109/ASONAM.2016.7752287), URL: <https://doi.org/10.1109/ASONAM.2016.7752287>.
- [236] Jonas Mueller and Aditya Thyagarajan, « Siamese Recurrent Architectures for Learning Sentence Similarity », *in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, Phoenix, Arizona: AAAI Press, 2016*, 2786–2792.
- [237] Marius Muja and David G. Lowe, « Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration », *in: VISAPP 2009 - Proceedings of the Fourth International Conference on Computer Vision Theory and Applications, Lisboa, Portugal, February 5-8, 2009 - Volume 1*, ed. by Alpesh Ranchordas and Helder Araújo, INSTICC Press, 2009, pp. 331–340.
- [238] Kai Nakamura, Sharon Levy, and William Yang Wang, « Fakeddit: A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection », *in: Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, ed. by Nicoletta Calzolari et al., European Language Resources Association, 2020, pp. 6149–6157, URL: <https://aclanthology.org/2020.lrec-1.755/>.
- [239] Ramesh Nallapati et al., « Abstractive text summarization using sequence-to-sequence rnns and beyond », *in: arXiv preprint arXiv:1602.06023* (2016).

-
- [240] Renato Negrinho, Matthew Gormley, and Geoffrey J Gordon, « Learning beam search policies via imitation learning », in: *Advances in Neural Information Processing Systems*, 2018, pp. 10652–10661.
- [241] Andrew Y. Ng and Michael I. Jordan, « On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes », in: *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, ed. by Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, MIT Press, 2001, pp. 841–848, URL: <https://proceedings.neurips.cc/paper/2001/hash/7b7a53e239400a13bd6be6c91c4f6c4e-Abstract.html>.
- [242] Jiquan Ngiam et al., « Multimodal Deep Learning », in: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, ed. by Lise Getoor and Tobias Scheffer, Omnipress, 2011, pp. 689–696, URL: https://icml.cc/2011/papers/399_icmlpaper.pdf.
- [243] Thao Nguyen et al., *Improving Multimodal Datasets with Image Captioning*, 2023, arXiv: 2307.10350 [cs.LG].
- [244] Maximilian Nickel et al., « A Review of Relational Machine Learning for Knowledge Graphs », in: *Proc. IEEE* 104.1 (2016), pp. 11–33, DOI: 10.1109/JPROC.2015.2483592, URL: <https://doi.org/10.1109/JPROC.2015.2483592>.
- [245] Raymond S. Nickerson, « Confirmation Bias: A Ubiquitous Phenomenon in Many Guises », in: *Review of General Psychology* 2.2 (1998), pp. 175–220, DOI: 10.1037/1089-2680.2.2.175, eprint: <https://doi.org/10.1037/1089-2680.2.2.175>, URL: <https://doi.org/10.1037/1089-2680.2.2.175>.
- [246] Mohammad Norouzi et al., « Reward Augmented Maximum Likelihood for Neural Structured Prediction », in: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, ed. by Daniel D. Lee et al., 2016, pp. 1723–1731, URL: <https://proceedings.neurips.cc/paper/2016/hash/2f885d0fbe2e131bfc9d98363e55d1d4-Abstract.html>.
- [247] Jekaterina Novikova et al., « Why We Need New Evaluation Metrics for NLG », in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, ed. by Martha

-
- Palmer, Rebecca Hwa, and Sebastian Riedel, Association for Computational Linguistics, 2017, pp. 2241–2252, DOI: [10.18653/v1/d17-1238](https://doi.org/10.18653/v1/d17-1238), URL: <https://doi.org/10.18653/v1/d17-1238>.
- [248] Aäron van den Oord, Yazhe Li, and Oriol Vinyals, « Representation Learning with Contrastive Predictive Coding », in: *CoRR* abs/1807.03748 (2018), arXiv: [1807.03748](https://arxiv.org/abs/1807.03748), URL: <http://arxiv.org/abs/1807.03748>.
- [249] OpenAI, « GPT-4 Technical Report », in: *CoRR* abs/2303.08774 (2023), DOI: [10.48550/arXiv.2303.08774](https://doi.org/10.48550/arXiv.2303.08774), arXiv: [2303.08774](https://arxiv.org/abs/2303.08774), URL: <https://doi.org/10.48550/arXiv.2303.08774>.
- [250] Shirui Pan et al., « Unifying Large Language Models and Knowledge Graphs: A Roadmap », in: *CoRR* abs/2306.08302 (2023), DOI: [10.48550/arXiv.2306.08302](https://doi.org/10.48550/arXiv.2306.08302), arXiv: [2306.08302](https://arxiv.org/abs/2306.08302), URL: <https://doi.org/10.48550/arXiv.2306.08302>.
- [251] Sinno Jialin Pan and Qiang Yang, « A Survey on Transfer Learning », in: *IEEE Trans. Knowl. Data Eng.* 22.10 (2010), pp. 1345–1359, DOI: [10.1109/TKDE.2009.191](https://doi.org/10.1109/TKDE.2009.191), URL: <https://doi.org/10.1109/TKDE.2009.191>.
- [252] Yannis Papanikolaou and Andrea Pierleoni, *DARE: Data Augmented Relation Extraction with GPT-2*, 2020, arXiv: [2004.13845](https://arxiv.org/abs/2004.13845) [cs.CL].
- [253] Nicolas Papernot et al., « The Limitations of Deep Learning in Adversarial Settings », in: *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, IEEE, 2016, pp. 372–387, DOI: [10.1109/EuroSP.2016.36](https://doi.org/10.1109/EuroSP.2016.36), URL: <https://doi.org/10.1109/EuroSP.2016.36>.
- [254] Kishore Papineni et al., « Bleu: a Method for Automatic Evaluation of Machine Translation », in: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, ACL, 2002, pp. 311–318, DOI: [10.3115/1073083.1073135](https://doi.org/10.3115/1073083.1073135), URL: <https://aclanthology.org/P02-1040/>.
- [255] Romain Paulus, Caiming Xiong, and Richard Socher, « A Deep Reinforced Model for Abstractive Summarization », in: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018, URL: <https://openreview.net/forum?id=HkAClQgA->.
- [256] F. Pedregosa et al., « Scikit-learn: Machine Learning in Python », in: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

-
- [257] Zhiliang Peng et al., « BEiT v2: Masked Image Modeling with Vector-Quantized Visual Tokenizers », in: *CoRR* abs/2208.06366 (2022), DOI: [10.48550/arXiv.2208.06366](https://doi.org/10.48550/arXiv.2208.06366), arXiv: [2208.06366](https://arxiv.org/abs/2208.06366), URL: <https://doi.org/10.48550/arXiv.2208.06366>.
- [258] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, « Glove: Global Vectors for Word Representation », in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans, ACL, 2014, pp. 1532–1543, DOI: [10.3115/v1/d14-1162](https://doi.org/10.3115/v1/d14-1162), URL: <https://doi.org/10.3115/v1/d14-1162>.
- [259] Verónica Pérez-Rosas et al., « Automatic Detection of Fake News », in: *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, ed. by Emily M. Bender, Leon Derczynski, and Pierre Isabelle, Association for Computational Linguistics, 2018, pp. 3391–3401, URL: <https://aclanthology.org/C18-1287/>.
- [260] Matthew E. Peters, Sebastian Ruder, and Noah A. Smith, « To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks », in: *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2019, Florence, Italy, August 2, 2019*, ed. by Isabelle Augenstein et al., Association for Computational Linguistics, 2019, pp. 7–14, DOI: [10.18653/v1/w19-4302](https://doi.org/10.18653/v1/w19-4302), URL: <https://doi.org/10.18653/v1/w19-4302>.
- [261] Matthew E. Peters et al., « Deep Contextualized Word Representations », in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, ed. by Marilyn A. Walker, Heng Ji, and Amanda Stent, Association for Computational Linguistics, 2018, pp. 2227–2237, DOI: [10.18653/v1/n18-1202](https://doi.org/10.18653/v1/n18-1202), URL: <https://doi.org/10.18653/v1/n18-1202>.
- [262] Vitali Petsiuk, Abir Das, and Kate Saenko, « RISE: Randomized Input Sampling for Explanation of Black-box Models », in: *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, BMVA Press, 2018, p. 151, URL: <http://bmvc2018.org/contents/papers/1064.pdf>.
- [263] Jason Phang, Thibault Févry, and Samuel R. Bowman, « Sentence Encoders on STILTs: Supplementary Training on Intermediate Labeled-data Tasks », in: *CoRR*

-
- abs/1811.01088 (2018), arXiv: [1811.01088](https://arxiv.org/abs/1811.01088), URL: <http://arxiv.org/abs/1811.01088>.
- [264] Konstantin Pogorelov et al., « FakeNews: Corona Virus and 5G Conspiracy Task at MediaEval 2020 », in: *MediaEval 2020 Workshop*, 2020.
- [265] Kashyap Popat, « Assessing the Credibility of Claims on the Web », in: *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, ed. by Rick Barrett et al., ACM, 2017, pp. 735–739, DOI: [10.1145/3041021.3053379](https://doi.org/10.1145/3041021.3053379), URL: <https://doi.org/10.1145/3041021.3053379>.
- [266] Peter Prettenhofer and Benno Stein, « Cross-Language Text Classification Using Structural Correspondence Learning », in: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 1118–1127, URL: <https://www.aclweb.org/anthology/P10-1114>.
- [267] Xiao Pu et al., *On the Zero-Shot Generalization of Machine-Generated Text Detectors*, 2023, arXiv: [2310.05165](https://arxiv.org/abs/2310.05165) [cs.CL].
- [268] Alec Radford et al., « Improving Language Understanding by Generative Pre-Training », in: 2018.
- [269] Alec Radford et al., « Language Models are Unsupervised Multitask Learners », in: *OpenAI Blog* (2019).
- [270] Alec Radford et al., « Learning Transferable Visual Models From Natural Language Supervision », in: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ed. by Marina Meila and Tong Zhang, vol. 139, Proceedings of Machine Learning Research, PMLR, 2021, pp. 8748–8763, URL: <http://proceedings.mlr.press/v139/radford21a.html>.
- [271] Colin Raffel et al., « Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer », in: *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67, URL: <http://jmlr.org/papers/v21/20-074.html>.
- [272] Pranav Rajpurkar et al., « SQuAD: 100,000+ Questions for Machine Comprehension of Text », in: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2383–2392.

-
- [273] Aditya Ramesh et al., « Zero-Shot Text-to-Image Generation », in: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ed. by Marina Meila and Tong Zhang, vol. 139, Proceedings of Machine Learning Research, PMLR, 2021, pp. 8821–8831, URL: <http://proceedings.mlr.press/v139/ramesh21a.html>.
- [274] Marc'Aurelio Ranzato et al., « Sequence Level Training with Recurrent Neural Networks », in: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, ed. by Yoshua Bengio and Yann LeCun, 2016, URL: <http://arxiv.org/abs/1511.06732>.
- [275] Benjamin Recht et al., « Do ImageNet Classifiers Generalize to ImageNet? », in: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov, vol. 97, Proceedings of Machine Learning Research, PMLR, 2019, pp. 5389–5400, URL: <http://proceedings.mlr.press/v97/recht19a.html>.
- [276] Shaoqing Ren et al., « Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks », in: *IEEE Trans. Pattern Anal. Mach. Intell.* 39.6 (2017), pp. 1137–1149, DOI: [10.1109/TPAMI.2016.2577031](https://doi.org/10.1109/TPAMI.2016.2577031), URL: <https://doi.org/10.1109/TPAMI.2016.2577031>.
- [277] Steven J. Rennie et al., « Self-Critical Sequence Training for Image Captioning », in: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, IEEE Computer Society, 2017, pp. 1179–1195, DOI: [10.1109/CVPR.2017.131](https://doi.org/10.1109/CVPR.2017.131), URL: <https://doi.org/10.1109/CVPR.2017.131>.
- [278] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin, « "Why Should I Trust You?": Explaining the Predictions of Any Classifier », in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, ed. by Balaji Krishnapuram et al., ACM, 2016, pp. 1135–1144, DOI: [10.1145/2939672.2939778](https://doi.org/10.1145/2939672.2939778), URL: <https://doi.org/10.1145/2939672.2939778>.
- [279] Marcel Robeer, Floris Bex, and Ad Feelders, « Generating Realistic Natural Language Counterfactuals », in: *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, As-

-
- sociation for Computational Linguistics, 2021, pp. 3611–3625, URL: <https://doi.org/10.18653/v1/2021.findings-emnlp.306>.
- [280] Juan Rodriguez et al., « Cross-Domain Detection of GPT-2-Generated Technical Text », in: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, Association for Computational Linguistics, 2022, pp. 1213–1233, DOI: [10.18653/v1/2022.naacl-main.88](https://doi.org/10.18653/v1/2022.naacl-main.88), URL: <https://doi.org/10.18653/v1/2022.naacl-main.88>.
- [281] Robin Rombach et al., « High-Resolution Image Synthesis with Latent Diffusion Models », in: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, IEEE, 2022, pp. 10674–10685, DOI: [10.1109/CVPR52688.2022.01042](https://doi.org/10.1109/CVPR52688.2022.01042), URL: <https://doi.org/10.1109/CVPR52688.2022.01042>.
- [282] Christopher D. Rosin, « Multi-armed bandits with episode context », in: *Ann. Math. Artif. Intell.* 61.3 (2011), pp. 203–230, DOI: [10.1007/s10472-011-9258-6](https://doi.org/10.1007/s10472-011-9258-6), URL: <https://doi.org/10.1007/s10472-011-9258-6>.
- [283] Aurko Roy et al., « Efficient Content-Based Sparse Attention with Routing Transformers », in: *Trans. Assoc. Comput. Linguistics* 9 (2021), pp. 53–68, DOI: [10.1162/tacl_a_00353](https://doi.org/10.1162/tacl_a_00353), URL: https://doi.org/10.1162/tacl_a_00353.
- [284] Baptiste Rozière et al., *Code Llama: Open Foundation Models for Code*, 2023, arXiv: [2308.12950](https://arxiv.org/abs/2308.12950) [cs.CL].
- [285] Victoria L. Rubin, « On deception and deception detection: Content analysis of computer-mediated stated beliefs », in: *Navigating Streams in an Information Ecosystem - Proceedings of the 73rd ASIS&T Annual Meeting, ASIST 2010, Pittsburgh, PA, USA, October 22-27, 2010*, vol. 47, Proc. Assoc. Inf. Sci. Technol. 1, Wiley, 2010, pp. 1–10, DOI: [10.1002/meet.14504701124](https://doi.org/10.1002/meet.14504701124), URL: <https://doi.org/10.1002/meet.14504701124>.
- [286] Victoria L. Rubin, Yimin Chen, and Nadia K. Conroy, « Deception detection for news: Three types of fakes », in: *Proceedings of the Association for Information Science and Technology* 52.1 (2015), pp. 1–4, DOI: <https://doi.org/10.1002/pa2.2015.145052010083>, eprint: <https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/pa2.2015.145052010083>.

-
- 10.1002/pra2.2015.145052010083, URL: <https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/pra2.2015.145052010083>.
- [287] Natali Ruchansky, Sungyong Seo, and Yan Liu, « CSI: A Hybrid Deep Model for Fake News Detection », in: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, ed. by Ee-Peng Lim et al., ACM, 2017, pp. 797–806, DOI: [10.1145/3132847.3132877](https://doi.org/10.1145/3132847.3132877), URL: <https://doi.org/10.1145/3132847.3132877>.
- [288] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, « Learning Internal Representations by Error Propagation », in: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition 1* (1986), 318–362.
- [289] Candida S. Punla et al., « Are we there yet?: An analysis of the competencies of BEED graduates of BPSU-DC », in: *International Multidisciplinary Research Journal* 4.3 (Sept. 2022), pp. 50–59.
- [290] Ekraam Sabir et al., « Deep Multimodal Image-Repurposing Detection », in: *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018*, ed. by Susanne Boll et al., ACM, 2018, pp. 1337–1345, DOI: [10.1145/3240508.3240707](https://doi.org/10.1145/3240508.3240707), URL: <https://doi.org/10.1145/3240508.3240707>.
- [291] Vinu Sankar Sadasivan et al., « Can AI-Generated Text be Reliably Detected? », in: *CoRR* abs/2303.11156 (2023), DOI: [10.48550/arXiv.2303.11156](https://doi.org/10.48550/arXiv.2303.11156), arXiv: [2303.11156](https://arxiv.org/abs/2303.11156), URL: <https://doi.org/10.48550/arXiv.2303.11156>.
- [292] Emmanuelle Salin et al., « Are Vision-Language Transformers Learning Multimodal Representations? A Probing Perspective », in: *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, AAAI Press, 2022, pp. 11248–11257, URL: <https://ojs.aaai.org/index.php/AAAI/article/view/21375>.
- [293] Guillaume Sanchez et al., « Stay on topic with Classifier-Free Guidance », in: *CoRR* abs/2306.17806 (2023), DOI: [10.48550/arXiv.2306.17806](https://doi.org/10.48550/arXiv.2306.17806), arXiv: [2306.17806](https://arxiv.org/abs/2306.17806), URL: <https://doi.org/10.48550/arXiv.2306.17806>.

-
- [294] Victor Sanh et al., « Multitask Prompted Training Enables Zero-Shot Task Generalization », in: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022, URL: <https://openreview.net/forum?id=9Vrb9D0WI4>.
- [295] Michael Santacrose et al., *Efficient RLHF: Reducing the Memory Usage of PPO*, 2023, arXiv: [2309.00754](https://arxiv.org/abs/2309.00754) [cs.LG].
- [296] Giovanni C. Santia and Jake Ryland Williams, « BuzzFace: A News Veracity Dataset with Facebook User Commentary and Egos », in: *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018*, AAAI Press, 2018, pp. 531–540, URL: <https://aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/view/17825>.
- [297] Elvis Saravia et al., « CARER: Contextualized Affect Representations for Emotion Recognition », in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, ed. by Ellen Riloff et al., Association for Computational Linguistics, 2018, pp. 3687–3697, DOI: [10.18653/v1/d18-1404](https://doi.org/10.18653/v1/d18-1404), URL: <https://doi.org/10.18653/v1/d18-1404>.
- [298] Daniel Thilo Schroeder, Konstantin Pogorelov, and Johannes Langguth, « FACT: a Framework for Analysis and Capture of Twitter Graphs », in: *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, IEEE, 2019, pp. 134–141.
- [299] Florian Schroff, Dmitry Kalenichenko, and James Philbin, « FaceNet: A unified embedding for face recognition and clustering », in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, IEEE Computer Society, 2015, pp. 815–823, DOI: [10.1109/CVPR.2015.7298682](https://doi.org/10.1109/CVPR.2015.7298682), URL: <https://doi.org/10.1109/CVPR.2015.7298682>.
- [300] Christoph Schuhmann et al., « LAION-5B: An open large-scale dataset for training next generation image-text models », in: *CoRR* abs/2210.08402 (2022), DOI: [10.48550/arXiv.2210.08402](https://doi.org/10.48550/arXiv.2210.08402), arXiv: [2210.08402](https://arxiv.org/abs/2210.08402), URL: <https://doi.org/10.48550/arXiv.2210.08402>.
- [301] Thomas Scialom et al., « Answers Unite! Unsupervised Metrics for Reinforced Summarization Models », in: *CoRR* abs/1909.01610 (2019), arXiv: [1909.01610](https://arxiv.org/abs/1909.01610), URL: <http://arxiv.org/abs/1909.01610>.

-
- [302] Thomas Scialom et al., « Coldgans: Taming language gans with cautious sampling strategies », in: *Advances in Neural Information Processing Systems* (2020).
- [303] Thomas Scialom et al., « Discriminative Adversarial Search for Abstractive Summarization », in: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, vol. 119, Proceedings of Machine Learning Research, PMLR, 2020, pp. 8555–8564, URL: <http://proceedings.mlr.press/v119/scialom20a.html>.
- [304] Thomas Scialom et al., « To Beam Or Not To Beam: That is a Question of Cooperation for Language GANs », in: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, ed. by Marc'Aurelio Ranzato et al., 2021, pp. 26585–26597, URL: <https://proceedings.neurips.cc/paper/2021/hash/df9028fcb6b065e000ffe8a4f03eeb38-Abstract.html>.
- [305] Thomas Scialom et al., « To Beam Or Not To Beam: That is a Question of Cooperation for Language GANs », in: *CoRR* abs/2106.06363 (2021), arXiv: [2106.06363](https://arxiv.org/abs/2106.06363), URL: <https://arxiv.org/abs/2106.06363>.
- [306] Abigail See et al., « Do Massively Pretrained Language Models Make Better Storytellers? », in: *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, ed. by Mohit Bansal and Aline Villavicencio, Association for Computational Linguistics, 2019, pp. 843–861, DOI: [10.18653/v1/K19-1079](https://doi.org/10.18653/v1/K19-1079), URL: <https://doi.org/10.18653/v1/K19-1079>.
- [307] Stanislaw Semeniuta, Aliaksei Severyn, and Sylvain Gelly, « On Accurate Evaluation of GANs for Language Generation », in: *CoRR* abs/1806.04936 (2018), arXiv: [1806.04936](https://arxiv.org/abs/1806.04936), URL: <http://arxiv.org/abs/1806.04936>.
- [308] Sharath M. Shankaranarayana and Davor Runje, « ALIME: Autoencoder Based Approach for Local Interpretability », in: *CoRR* abs/1909.02437 (2019), URL: <http://arxiv.org/abs/1909.02437>.
- [309] Chengcheng Shao et al., « The spread of low-credibility content by social bots », in: *Nature Communications* 9 (2018), p. 4787, DOI: [10.1038/s41467-018-06930-7](https://doi.org/10.1038/s41467-018-06930-7), URL: <https://doi.org/10.1038/s41467-018-06930-7>.

-
- [310] Piyush Sharma et al., « Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning », in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, ed. by Iryna Gurevych and Yusuke Miyao, Association for Computational Linguistics, 2018, pp. 2556–2565, DOI: [10.18653/v1/P18-1238](https://doi.org/10.18653/v1/P18-1238), URL: <https://aclanthology.org/P18-1238/>.
- [311] Baoxu Shi and Tim Weninger, « Discriminative predicate path mining for fact checking in knowledge graphs », in: *Knowl. Based Syst.* 104 (2016), pp. 123–133, DOI: [10.1016/j.knosys.2016.04.015](https://doi.org/10.1016/j.knosys.2016.04.015), URL: <https://doi.org/10.1016/j.knosys.2016.04.015>.
- [312] Connor Shorten and Taghi Khoshgoftaar, « A survey on Image Data Augmentation for Deep Learning », in: *Journal of Big Data* 6 (July 2019), DOI: [10.1186/s40537-019-0197-0](https://doi.org/10.1186/s40537-019-0197-0).
- [313] Kai Shu, H. Russell Bernard, and Huan Liu, « Studying Fake News via Network Analysis: Detection and Mitigation », in: *CoRR* abs/1804.10233 (2018), arXiv: [1804.10233](https://arxiv.org/abs/1804.10233), URL: <http://arxiv.org/abs/1804.10233>.
- [314] Kai Shu, Suhang Wang, and Huan Liu, « Beyond News Contents: The Role of Social Context for Fake News Detection », in: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, ed. by J. Shane Culpepper et al., ACM, 2019, pp. 312–320, DOI: [10.1145/3289600.3290994](https://doi.org/10.1145/3289600.3290994), URL: <https://doi.org/10.1145/3289600.3290994>.
- [315] Kai Shu et al., « FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media », in: *Big Data* 8.3 (2020), pp. 171–188, DOI: [10.1089/big.2020.0062](https://doi.org/10.1089/big.2020.0062), URL: <https://doi.org/10.1089/big.2020.0062>.
- [316] David Silver et al., « A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play », in: *Science* 362.6419 (2018), pp. 1140–1144, ISSN: 0036-8075, DOI: [10.1126/science.aar6404](https://doi.org/10.1126/science.aar6404), eprint: <https://science.sciencemag.org/content/362/6419/1140.full.pdf>, URL: <https://science.sciencemag.org/content/362/6419/1140>.

-
- [317] David Silver et al., « Mastering the game of Go without human knowledge », in: *Nat.* 550.7676 (2017), pp. 354–359, DOI: [10.1038/nature24270](https://doi.org/10.1038/nature24270), URL: <https://doi.org/10.1038/nature24270>.
- [318] Karen Simonyan and Andrew Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition », in: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, ed. by Yoshua Bengio and Yann LeCun, 2015, URL: <http://arxiv.org/abs/1409.1556>.
- [319] Uriel Singer et al., *Make-A-Video: Text-to-Video Generation without Text-Video Data*, 2022, arXiv: [2209.14792](https://arxiv.org/abs/2209.14792) [cs.CV].
- [320] Shivangi Singhal et al., « SpotFake: A Multi-modal Framework for Fake News Detection », in: *Fifth IEEE International Conference on Multimedia Big Data, BigMM 2019, Singapore, September 11-13, 2019*, IEEE, 2019, pp. 39–47, DOI: [10.1109/BigMM.2019.00-44](https://doi.org/10.1109/BigMM.2019.00-44), URL: <https://doi.org/10.1109/BigMM.2019.00-44>.
- [321] Shivangi Singhal et al., « SpotFake+: A Multimodal Framework for Fake News Detection via Transfer Learning (Student Abstract) », in: *Proceedings of the AAAI Conference on Artificial Intelligence 34.10* (Apr. 2020), pp. 13915–13916, DOI: [10.1609/aaai.v34i10.7230](https://doi.org/10.1609/aaai.v34i10.7230), URL: <https://ojs.aaai.org/index.php/AAAI/article/view/7230>.
- [322] Niraj Sitaula et al., « Credibility-based Fake News Detection », in: *CoRR* abs/1911.00643 (2019), arXiv: [1911.00643](https://arxiv.org/abs/1911.00643), URL: <http://arxiv.org/abs/1911.00643>.
- [323] Josef Sivic and Andrew Zisserman, « Video Google: A Text Retrieval Approach to Object Matching in Videos », in: *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, IEEE Computer Society, 2003, pp. 1470–1477, DOI: [10.1109/ICCV.2003.1238663](https://doi.org/10.1109/ICCV.2003.1238663), URL: <https://doi.org/10.1109/ICCV.2003.1238663>.
- [324] Kihyuk Sohn, « Improved Deep Metric Learning with Multi-class N-pair Loss Objective », in: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, ed. by Daniel D. Lee et al., 2016, pp. 1849–1857, URL: <https://proceedings.neurips.cc/paper/2016/hash/6b180037abbebea991d8b1232f8a8ca9-Abstract.html>.

-
- [325] Irene Solaiman et al., « Release Strategies and the Social Impacts of Language Models », in: *CoRR* abs/1908.09203 (2019), arXiv: [1908.09203](https://arxiv.org/abs/1908.09203), URL: <http://arxiv.org/abs/1908.09203>.
- [326] Nitish Srivastava et al., « Dropout: a simple way to prevent neural networks from overfitting », in: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1929–1958, DOI: [10.5555/2627435.2670313](https://doi.org/10.5555/2627435.2670313), URL: <https://dl.acm.org/doi/10.5555/2627435.2670313>.
- [327] « Statistical Theory of Extreme Values and Some Practical Applications. Lectures by Emit J. Gumbel. National Bureau of Standards, Washington, 1954. 51 pp. Diagrams. 40 cents. », in: *The Aeronautical Journal* 58.527 (1954), 792–793, DOI: [10.1017/S0368393100099958](https://doi.org/10.1017/S0368393100099958).
- [328] Deryn Strange et al., « Photographs cause false memories for the news », in: *Acta Psychologica* 136.1 (2011), pp. 90–94, ISSN: 0001-6918, DOI: <https://doi.org/10.1016/j.actpsy.2010.10.006>, URL: <https://www.sciencedirect.com/science/article/pii/S000169181000212X>.
- [329] Weijie Su et al., « VL-BERT: Pre-training of Generic Visual-Linguistic Representations », in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020, URL: <https://openreview.net/forum?id=SygXPaEYvH>.
- [330] Alane Suhr et al., « A Corpus of Natural Language for Visual Reasoning », in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, ed. by Regina Barzilay and Min-Yen Kan, Association for Computational Linguistics, 2017, pp. 217–223, DOI: [10.18653/v1/P17-2034](https://doi.org/10.18653/v1/P17-2034), URL: <https://doi.org/10.18653/v1/P17-2034>.
- [331] Chi Sun et al., « How to Fine-Tune BERT for Text Classification? », in: *Chinese Computational Linguistics - 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings*, ed. by Maosong Sun et al., vol. 11856, Lecture Notes in Computer Science, Springer, 2019, pp. 194–206, DOI: [10.1007/978-3-030-32381-3_16](https://doi.org/10.1007/978-3-030-32381-3_16), URL: https://doi.org/10.1007/978-3-030-32381-3_16.
- [332] Quan Sun et al., *Generative Pretraining in Multimodality*, 2023, arXiv: [2307.05222](https://arxiv.org/abs/2307.05222) [cs.CV].

-
- [333] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, « Sequence to Sequence Learning with Neural Networks », in: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, ed. by Zoubin Ghahramani et al., 2014, pp. 3104–3112, URL: <https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html>.
- [334] Richard S. Sutton and Andrew G. Barto, *Reinforcement learning - an introduction*, Adaptive computation and machine learning, MIT Press, 1998, ISBN: 978-0-262-19398-6, URL: <https://www.worldcat.org/oclc/37293240>.
- [335] Christian Szegedy et al., « Going deeper with convolutions », in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [336] Eugenio Tacchini et al., « Some Like it Hoax: Automated Fake News Detection in Social Networks », in: *CoRR abs/1704.07506* (2017), arXiv: [1704.07506](https://arxiv.org/abs/1704.07506), URL: <http://arxiv.org/abs/1704.07506>.
- [337] Andrea Tagarelli and Andrea Simeri, « Unsupervised law article mining based on deep pre-trained language representation models with application to the Italian civil code », in: *Artificial Intelligence and Law 30.3* (Sept. 2022), pp. 417–473, ISSN: 1572-8382, URL: <https://doi.org/10.1007/s10506-021-09301-8>.
- [338] Hao Tan and Mohit Bansal, « LXMERT: Learning Cross-Modality Encoder Representations from Transformers », in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, ed. by Kentaro Inui et al., Association for Computational Linguistics, 2019, pp. 5099–5110, DOI: [10.18653/v1/D19-1514](https://doi.org/10.18653/v1/D19-1514), URL: <https://doi.org/10.18653/v1/D19-1514>.
- [339] Reuben Tan, Bryan A. Plummer, and Kate Saenko, « Detecting Cross-Modal Inconsistency to Defend Against Neural Fake News », in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, ed. by Bonnie Webber et al., Association for Computational Linguistics, 2020, pp. 2081–2106, DOI: [10.18653/v1/2020.emnlp-main.163](https://doi.org/10.18653/v1/2020.emnlp-main.163), URL: <https://doi.org/10.18653/v1/2020.emnlp-main.163>.
- [340] Rohan Taori et al., *Alpaca: A Strong, Replicable Instruction-Following Model*, <https://crfm.stanford.edu/2023/03/13/alpaca.html>, 2023.

-
- [341] Rohan Taori et al., « Measuring Robustness to Natural Distribution Shifts in Image Classification », in: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, ed. by Hugo Larochelle et al., 2020, URL: <https://proceedings.neurips.cc/paper/2020/hash/d8330f857a17c53d217014ee776bfd50-Abstract.html>.
- [342] Yi Tay et al., « Long Range Arena : A Benchmark for Efficient Transformers », in: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021, URL: <https://openreview.net/forum?id=qVyeW-grC2k>.
- [343] Yi Tay et al., « Reverse Engineering Configurations of Neural Text Generation Models », in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, ed. by Dan Jurafsky et al., Association for Computational Linguistics, 2020, pp. 275–279, DOI: [10.18653/v1/2020.acl-main.25](https://doi.org/10.18653/v1/2020.acl-main.25), URL: <https://doi.org/10.18653/v1/2020.acl-main.25>.
- [344] Wilson L. Taylor, « “Cloze Procedure”: A New Tool for Measuring Readability », in: *Journalism & Mass Communication Quarterly* 30 (1953), pp. 415–433.
- [345] Ian Tenney, Dipanjan Das, and Ellie Pavlick, « BERT Rediscovered the Classical NLP Pipeline », in: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, ed. by Anna Korhonen, David R. Traum, and Lluís Màrquez, Association for Computational Linguistics, 2019, pp. 4593–4601, DOI: [10.18653/v1/p19-1452](https://doi.org/10.18653/v1/p19-1452), URL: <https://doi.org/10.18653/v1/p19-1452>.
- [346] Guy Tevet et al., « Evaluating Text GANs as Language Models », in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, ed. by Jill Burstein, Christy Doran, and Thamar Solorio, Association for Computational Linguistics, 2019, pp. 2241–2247, DOI: [10.18653/v1/n19-1233](https://doi.org/10.18653/v1/n19-1233), URL: <https://doi.org/10.18653/v1/n19-1233>.
- [347] Hugo Touvron et al., *Llama 2: Open Foundation and Fine-Tuned Chat Models*, 2023, arXiv: [2307.09288](https://arxiv.org/abs/2307.09288) [cs.CL].

-
- [348] Michael Tschannen et al., « Image Captioners Are Scalable Vision Learners Too », in: *CoRR* abs/2306.07915 (2023), DOI: [10.48550/arXiv.2306.07915](https://doi.org/10.48550/arXiv.2306.07915), arXiv: [2306.07915](https://doi.org/10.48550/arXiv.2306.07915), URL: <https://doi.org/10.48550/arXiv.2306.07915>.
- [349] A. M. TURING, « I.—COMPUTING MACHINERY AND INTELLIGENCE », in: *Mind* LIX.236 (Oct. 1950), pp. 433–460, ISSN: 0026-4423, DOI: [10.1093/mind/LIX.236.433](https://doi.org/10.1093/mind/LIX.236.433), eprint: <https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf>, URL: <https://doi.org/10.1093/mind/LIX.236.433>.
- [350] Adaku Uchendu et al., « Authorship Attribution for Neural Text Generation », in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, ed. by Bonnie Webber et al., Association for Computational Linguistics, 2020, pp. 8384–8395, DOI: [10.18653/v1/2020.emnlp-main.673](https://doi.org/10.18653/v1/2020.emnlp-main.673), URL: <https://doi.org/10.18653/v1/2020.emnlp-main.673>.
- [351] Onur Varol et al., « Online Human-Bot Interactions: Detection, Estimation, and Characterization », in: *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017*, AAAI Press, 2017, pp. 280–289, URL: <https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15587>.
- [352] Lav R. Varshney, Nitish Shirish Keskar, and Richard Socher, « Limits of Detecting Text Generated by Large-Scale Language Models », in: *Information Theory and Applications Workshop, ITA 2020, San Diego, CA, USA, February 2-7, 2020*, IEEE, 2020, pp. 1–5, DOI: [10.1109/ITA50056.2020.9245012](https://doi.org/10.1109/ITA50056.2020.9245012), URL: <https://doi.org/10.1109/ITA50056.2020.9245012>.
- [353] Ashish Vaswani et al., « Attention is All you Need », in: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, ed. by Isabelle Guyon et al., 2017, pp. 5998–6008, URL: <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>.
- [354] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh, « CIDEr: Consensus-based image description evaluation », in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, IEEE Computer Society, 2015, pp. 4566–4575, DOI: [10.1109/CVPR.2015.7299087](https://doi.org/10.1109/CVPR.2015.7299087), URL: <https://doi.org/10.1109/CVPR.2015.7299087>.

-
- [355] Gilad Vered et al., « Joint Optimization for Cooperative Image Captioning », in: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8897–8906, DOI: [10.1109/ICCV.2019.00899](https://doi.org/10.1109/ICCV.2019.00899).
- [356] Giorgio Visani, Enrico Bagli, and Federico Chesani, « OptiLIME: Optimized LIME Explanations for Diagnostic Computer Algorithms », in: *Proceedings of the CIKM 2020 Workshops co-located with 29th ACM International Conference on Information and Knowledge Management (CIKM 2020), Galway, Ireland, October 19-23, 2020*, vol. 2699, CEUR Workshop Proceedings, CEUR-WS.org, 2020, URL: <http://ceur-ws.org/Vol-2699/paper03.pdf>.
- [357] Marco Viviani and Gabriella Pasi, « Credibility in social media: opinions, news, and health information—a survey », in: *WIREs Data Mining and Knowledge Discovery 7.5* (2017), e1209, DOI: <https://doi.org/10.1002/widm.1209>, eprint: <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1209>, URL: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1209>.
- [358] Nguyen Vo and Kyumin Lee, « The Rise of Guardians: Fact-checking URL Recommendation to Combat Fake News », in: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, ed. by Kevyn Collins-Thompson et al., ACM, 2018, pp. 275–284, DOI: [10.1145/3209978.3210037](https://doi.org/10.1145/3209978.3210037), URL: <https://doi.org/10.1145/3209978.3210037>.
- [359] Soroush Vosoughi, Deb Roy, and Sinan Aral, « The spread of true and false news online », in: *Science 359.6380* (2018), pp. 1146–1151, DOI: [10.1126/science.aap9559](https://doi.org/10.1126/science.aap9559), eprint: <https://www.science.org/doi/pdf/10.1126/science.aap9559>, URL: <https://www.science.org/doi/abs/10.1126/science.aap9559>.
- [360] Vedran Vukotić, Vivien Chappelier, and Teddy Furon, « Are Deep Neural Networks good for blind image watermarking? », in: *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018, pp. 1–7, DOI: [10.1109/WIFS.2018.8630768](https://doi.org/10.1109/WIFS.2018.8630768).
- [361] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell, « Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR », in: *CoRR* (2017), arXiv: [1711.00399](https://arxiv.org/abs/1711.00399), URL: <http://arxiv.org/abs/1711.00399>.
- [362] Chengyi Wang et al., *Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers*, 2023, arXiv: [2301.02111](https://arxiv.org/abs/2301.02111) [cs.CL].

-
- [363] Feng Wang and Huaping Liu, « Understanding the Behaviour of Contrastive Loss », in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, Computer Vision Foundation / IEEE, 2021, pp. 2495–2504, DOI: [10.1109/CVPR46437.2021.00252](https://doi.org/10.1109/CVPR46437.2021.00252), URL: https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Understanding_the_Behaviour_of_Contrastive_Loss_CVPR_2021_paper.html.
- [364] Peng Wang et al., « OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework », in: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ed. by Kamalika Chaudhuri et al., vol. 162, Proceedings of Machine Learning Research, PMLR, 2022, pp. 23318–23340, URL: <https://proceedings.mlr.press/v162/wang22a1.html>.
- [365] Qingzhong Wang and Antoni B. Chan, « Describing Like Humans: On Diversity in Image Captioning », in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, 2019, pp. 4195–4203, DOI: [10.1109/CVPR.2019.00432](https://doi.org/10.1109/CVPR.2019.00432), URL: http://openaccess.thecvf.com/content_CVPR_2019/html/Wang_Describing_Like_Humans_On_Diversity_in_Image_Captioning_CVPR_2019_paper.html.
- [366] Sinong Wang et al., « Linformer: Self-Attention with Linear Complexity », in: *CoRR abs/2006.04768 (2020)*, arXiv: [2006.04768](https://arxiv.org/abs/2006.04768), URL: <https://arxiv.org/abs/2006.04768>.
- [367] Thomas Wang et al., « What Language Model Architecture and Pretraining Objective Works Best for Zero-Shot Generalization? », in: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ed. by Kamalika Chaudhuri et al., vol. 162, Proceedings of Machine Learning Research, PMLR, 2022, pp. 22964–22984, URL: <https://proceedings.mlr.press/v162/wang22u.html>.
- [368] Wenhui Wang et al., « Image as a Foreign Language: BEiT Pretraining for All Vision and Vision-Language Tasks », in: *CoRR abs/2208.10442 (2022)*, DOI: [10.48550/arXiv.2208.10442](https://doi.org/10.48550/arXiv.2208.10442), arXiv: [2208.10442](https://arxiv.org/abs/2208.10442), URL: <https://doi.org/10.48550/arXiv.2208.10442>.

-
- [369] William Yang Wang, « "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection », in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, ed. by Regina Barzilay and Min-Yen Kan, Association for Computational Linguistics, 2017, pp. 422–426, DOI: [10.18653/v1/P17-2067](https://doi.org/10.18653/v1/P17-2067), URL: <https://doi.org/10.18653/v1/P17-2067>.
- [370] William Yang Wang and Diyi Yang, « That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets », in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 2557–2563, DOI: [10.18653/v1/D15-1306](https://doi.org/10.18653/v1/D15-1306), URL: <https://aclanthology.org/D15-1306>.
- [371] Xinyi Wang et al., « SwitchOut: an Efficient Data Augmentation Algorithm for Neural Machine Translation », in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 856–861, DOI: [10.18653/v1/D18-1100](https://doi.org/10.18653/v1/D18-1100), URL: <https://aclanthology.org/D18-1100>.
- [372] Yaqing Wang et al., « EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection », in: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, ed. by Yike Guo and Faisal Farooq, ACM, 2018, pp. 849–857, DOI: [10.1145/3219819.3219903](https://doi.org/10.1145/3219819.3219903), URL: <https://doi.org/10.1145/3219819.3219903>.
- [373] Zhuhao Wang et al., « Diverse Image Captioning via GroupTalk », in: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, ed. by Subbarao Kambhampati, IJCAI/AAAI Press, 2016, pp. 2957–2964, URL: <http://www.ijcai.org/Abstract/16/420>.
- [374] Zeerak Waseem et al., « Understanding Abuse: A Typology of Abusive Language Detection Subtasks », in: *Proceedings of the First Workshop on Abusive Language Online, ALW@ACL 2017, Vancouver, BC, Canada, August 4, 2017*, ed. by Zeerak Waseem et al., Association for Computational Linguistics, 2017, pp. 78–84, DOI: [10.18653/v1/w17-3012](https://doi.org/10.18653/v1/w17-3012), URL: <https://doi.org/10.18653/v1/w17-3012>.

-
- [375] Jason Wei and Kai Zou, « EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks », in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6382–6388, DOI: [10.18653/v1/D19-1670](https://doi.org/10.18653/v1/D19-1670), URL: <https://www.aclweb.org/anthology/D19-1670>.
- [376] Jason Wei et al., « Finetuned Language Models are Zero-Shot Learners », in: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022, URL: <https://openreview.net/forum?id=gEZrGCozdqR>.
- [377] Laura Weidinger et al., *Ethical and social risks of harm from Language Models*, 2021, arXiv: [2112.04359](https://arxiv.org/abs/2112.04359) [cs.CL].
- [378] Joseph Weizenbaum, « ELIZA a Computer Program for the Study of Natural Language Communication Between Man and Machine », in: *Commun. ACM* 9.1 (Jan. 1966), pp. 36–45, ISSN: 0001-0782, DOI: [10.1145/365153.365168](https://doi.org/10.1145/365153.365168), URL: <http://doi.acm.org/10.1145/365153.365168>.
- [379] Sean Welleck et al., « Neural text generation with unlikelihood training », in: *arXiv preprint arXiv:1908.04319* (2019).
- [380] Ronald J. Williams, « Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning », in: *Mach. Learn.* 8 (1992), pp. 229–256, DOI: [10.1007/BF00992696](https://doi.org/10.1007/BF00992696), URL: <https://doi.org/10.1007/BF00992696>.
- [381] Ronald J. Williams and David Zipser, « A Learning Algorithm for Continually Running Fully Recurrent Neural Networks », in: *Neural Comput.* 1.2 (1989), pp. 270–280, DOI: [10.1162/neco.1989.1.2.270](https://doi.org/10.1162/neco.1989.1.2.270), URL: <https://doi.org/10.1162/neco.1989.1.2.270>.
- [382] Thomas Wolf et al., « Transformers: State-of-the-Art Natural Language Processing », in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, ed. by Qun Liu and David Schlangen, Association for Computational Linguistics, 2020, pp. 38–45, DOI: [10.18653/v1/2020.emnlp-demos.6](https://doi.org/10.18653/v1/2020.emnlp-demos.6), URL: <https://doi.org/10.18653/v1/2020.emnlp-demos.6>.

-
- [383] Max Wolff and Stuart Wolff, « Attacking neural text detectors », in: *arXiv preprint arXiv:2002.11768* (2020).
- [384] Lin Wu et al., « Generating Life Course Trajectory Sequences with Recurrent Neural Networks and Application to Early Detection of Social Disadvantage », in: *Advanced Data Mining and Applications - 13th International Conference, ADMA 2017, Singapore, November 5-6, 2017, Proceedings*, ed. by Gao Cong et al., vol. 10604, Lecture Notes in Computer Science, Springer, 2017, pp. 225–242, DOI: [10.1007/978-3-319-69179-4_16](https://doi.org/10.1007/978-3-319-69179-4_16), URL: https://doi.org/10.1007/978-3-319-69179-4_16.
- [385] Tongshuang Wu et al., « Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models », in: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, ed. by Chengqing Zong et al., Association for Computational Linguistics, 2021, pp. 6707–6723, DOI: [10.18653/v1/2021.acl-long.523](https://doi.org/10.18653/v1/2021.acl-long.523), URL: <https://doi.org/10.18653/v1/2021.acl-long.523>.
- [386] Xing Wu et al., « Conditional BERT Contextual Augmentation », in: *Computational Science – ICCS 2019*, ed. by João M. F. Rodrigues et al., Cham: Springer International Publishing, 2019, pp. 84–95, ISBN: 978-3-030-22747-0.
- [387] Qizhe Xie et al., « Unsupervised Data Augmentation for Consistency Training », in: *Advances in Neural Information Processing Systems*, ed. by H. Larochelle et al., vol. 33, Curran Associates, Inc., 2020, pp. 6256–6268, URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/44feb0096faa8326192570788b38c1d1-Paper.pdf.
- [388] Saining Xie et al., *Aggregated Residual Transformations for Deep Neural Networks*, 2017, arXiv: [1611.05431](https://arxiv.org/abs/1611.05431) [cs.CV].
- [389] Jin Xu et al., « Learning to Break the Loop: Analyzing and Mitigating Repetitions for Neural Text Generation », in: *NeurIPS*, 2022, URL: http://papers.nips.cc/paper_files/paper/2022/hash/148c0aeea1c5da82f4fa86a09d4190da-Abstract-Conference.html.
- [390] Junxiao Xue et al., « Detecting fake news by exploring the consistency of multimodal data », in: *Information Processing & Management* 58.5 (2021), p. 102610, ISSN: 0306-4573, DOI: <https://doi.org/10.1016/j.ipm.2021.102610>, URL: <https://www.sciencedirect.com/science/article/pii/S0306457321001060>.

-
- [391] Ge Yan et al., « Data Augmentation for Deep Learning of Judgment Documents », in: *Intelligence Science and Big Data Engineering. Big Data and Machine Learning*, ed. by Zhen Cui et al., Cham: Springer International Publishing, 2019, pp. 232–242, ISBN: 978-3-030-36204-1.
- [392] Chun-Hsiao Yeh et al., « Decoupled Contrastive Learning », in: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVI*, ed. by Shai Avidan et al., vol. 13686, Lecture Notes in Computer Science, Springer, 2022, pp. 668–684, DOI: [10.1007/978-3-031-19809-0_38](https://doi.org/10.1007/978-3-031-19809-0_38), URL: https://doi.org/10.1007/978-3-031-19809-0_38.
- [393] Dani Yogatama et al., « Generative and Discriminative Text Classification with Recurrent Neural Networks », in: *CoRR* abs/1703.01898 (2017), arXiv: [1703.01898](https://arxiv.org/abs/1703.01898), URL: <http://arxiv.org/abs/1703.01898>.
- [394] Peter Young et al., « From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions », in: *Trans. Assoc. Comput. Linguistics* 2 (2014), pp. 67–78, DOI: [10.1162/tacl_a_00166](https://doi.org/10.1162/tacl_a_00166), URL: https://doi.org/10.1162/tacl_a_00166.
- [395] Adams Wei Yu et al., « Fast and Accurate Reading Comprehension by Combining Self-Attention and Convolution », in: *International Conference on Learning Representations*, 2018, URL: <https://openreview.net/forum?id=B14TlG-RW>.
- [396] Bowen Yu et al., « Beyond Word Attention: Using Segment Attention in Neural Relation Extraction », in: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, ed. by Sarit Kraus, ijcai.org, 2019, pp. 5401–5407, DOI: [10.24963/ijcai.2019/750](https://doi.org/10.24963/ijcai.2019/750), URL: <https://doi.org/10.24963/ijcai.2019/750>.
- [397] Jiahui Yu et al., « CoCa: Contrastive Captioners are Image-Text Foundation Models », in: *CoRR* abs/2205.01917 (2022), DOI: [10.48550/arXiv.2205.01917](https://doi.org/10.48550/arXiv.2205.01917), arXiv: [2205.01917](https://arxiv.org/abs/2205.01917), URL: <https://doi.org/10.48550/arXiv.2205.01917>.
- [398] Lantao Yu et al., « SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient », in: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, ed. by Satinder Singh and Shaul Markovitch, AAAI Press, 2017, pp. 2852–2858, URL: <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14344>.

-
- [399] Youngjae Yu et al., « Multimodal Knowledge Alignment with Reinforcement Learning », in: *CoRR* abs/2205.12630 (2022), DOI: [10.48550/arXiv.2205.12630](https://doi.org/10.48550/arXiv.2205.12630), arXiv: [2205.12630](https://arxiv.org/abs/2205.12630), URL: <https://doi.org/10.48550/arXiv.2205.12630>.
- [400] Ruifeng Yuan, Zili Wang, and Wenjie Li, « Event Graph based Sentence Fusion », in: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, ed. by Marie-Francine Moens et al., Association for Computational Linguistics, 2021, pp. 4075–4084, DOI: [10.18653/v1/2021.emnlp-main.334](https://doi.org/10.18653/v1/2021.emnlp-main.334), URL: <https://doi.org/10.18653/v1/2021.emnlp-main.334>.
- [401] Muhammad Rehman Zafar and Naimul Mefraz Khan, « DLIME: A Deterministic Local Interpretable Model-Agnostic Explanations Approach for Computer-Aided Diagnosis Systems », in: *CoRR* abs/1906.10263 (2019), URL: <http://arxiv.org/abs/1906.10263>.
- [402] Sergey Zagoruyko and Nikos Komodakis, *Wide Residual Networks*, 2017, arXiv: [1605.07146](https://arxiv.org/abs/1605.07146) [cs.CV].
- [403] Manzil Zaheer et al., « Big Bird: Transformers for Longer Sequences », in: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, ed. by Hugo Larochelle et al., 2020, URL: <https://proceedings.neurips.cc/paper/2020/hash/c8512d142a2d849725f31a9a7a361ab9-Abstract.html>.
- [404] Savvas Zannettou et al., « On the Origins of Memes by Means of Fringe Web Communities », in: *Proceedings of the Internet Measurement Conference 2018, IMC 2018, Boston, MA, USA, October 31 - November 02, 2018*, ACM, 2018, pp. 188–202, URL: <https://dl.acm.org/citation.cfm?id=3278550>.
- [405] Rowan Zellers et al., « Defending Against Neural Fake News », in: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, ed. by Hanna M. Wallach et al., 2019, pp. 9051–9062, URL: <https://proceedings.neurips.cc/paper/2019/hash/3e9f0fc9b2f89e043bc6233994dfcf76-Abstract.html>.

-
- [406] Rowan Zellers et al., « From Recognition to Cognition: Visual Commonsense Reasoning », in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, 2019, pp. 6720–6731, DOI: [10.1109/CVPR.2019.00688](https://doi.org/10.1109/CVPR.2019.00688), URL: http://openaccess.thecvf.com/content/_CVPR/_2019/html/Zellers_From_Recognition_to_Cognition_Visual_Commonsense_Reasoning_CVPR_2019_paper.html.
- [407] Pengchuan Zhang et al., « VinVL: Revisiting Visual Representations in Vision-Language Models », in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, Computer Vision Foundation / IEEE, 2021, pp. 5579–5588, DOI: [10.1109/CVPR46437.2021.00553](https://doi.org/10.1109/CVPR46437.2021.00553), URL: https://openaccess.thecvf.com/content/CVPR2021/html/Zhang_VinVL_Revisiting_Visual_Representations_in_Vision-Language_Models_CVPR_2021_paper.html.
- [408] Shun Zhang et al., « Planning with Large Language Models for Code Generation », in: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, OpenReview.net, 2023, URL: <https://openreview.net/pdf?id=Lr8c00tYbfL>.
- [409] Susan Zhang et al., « OPT: Open Pre-trained Transformer Language Models », in: *CoRR* abs/2205.01068 (2022), DOI: [10.48550/arXiv.2205.01068](https://doi.org/10.48550/arXiv.2205.01068), arXiv: [2205.01068](https://arxiv.org/abs/2205.01068), URL: <https://doi.org/10.48550/arXiv.2205.01068>.
- [410] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun, « Character-level Convolutional Networks for Text Classification », in: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, ed. by Corinna Cortes et al., 2015, pp. 649–657, URL: <https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html>.
- [411] Youyuan Zhang et al., « Distinctive Image Captioning via CLIP Guided Group Optimization », in: *Computer Vision - ECCV 2022 Workshops - Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IV*, ed. by Leonid Karlinsky, Tomer Michaeli, and Ko Nishino, vol. 13804, Lecture Notes in Computer Science, Springer, 2022, pp. 223–238, DOI: [10.1007/978-3-031-25069-9_15](https://doi.org/10.1007/978-3-031-25069-9_15), URL: https://doi.org/10.1007/978-3-031-25069-9_15.

-
- [412] Wayne Xin Zhao et al., « A Survey of Large Language Models », in: *CoRR* abs/2303.18223 (2023), DOI: [10.48550/arXiv.2303.18223](https://doi.org/10.48550/arXiv.2303.18223), arXiv: [2303.18223](https://arxiv.org/abs/2303.18223), URL: <https://doi.org/10.48550/arXiv.2303.18223>.
- [413] Xuandong Zhao et al., « Provable Robust Watermarking for AI-Generated Text », in: *CoRR* abs/2306.17439 (2023), DOI: [10.48550/arXiv.2306.17439](https://doi.org/10.48550/arXiv.2306.17439), arXiv: [2306.17439](https://arxiv.org/abs/2306.17439), URL: <https://doi.org/10.48550/arXiv.2306.17439>.
- [414] Xudong Zhao et al., « Detecting Digital Image Splicing in Chroma Spaces », in: *Digital Watermarking - 9th International Workshop, IWDW 2010, Seoul, Korea, October 1-3, 2010, Revised Selected Papers*, ed. by Hyoung-Joong Kim, Yun-Qing Shi, and Mauro Barni, vol. 6526, Lecture Notes in Computer Science, Springer, 2010, pp. 12–22, DOI: [10.1007/978-3-642-18405-5_2](https://doi.org/10.1007/978-3-642-18405-5_2), URL: https://doi.org/10.1007/978-3-642-18405-5_2.
- [415] Zilong Zhao et al., « Fake news propagates differently from real news even at early stages of spreading », in: *EPJ Data Sci.* 9.1 (2020), p. 7, DOI: [10.1140/epjds/s13688-020-00224-z](https://doi.org/10.1140/epjds/s13688-020-00224-z), URL: <https://doi.org/10.1140/epjds/s13688-020-00224-z>.
- [416] Peng Zhou et al., « Learning Rich Features for Image Manipulation Detection », in: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 1053–1061, DOI: [10.1109/CVPR.2018.00116](https://doi.org/10.1109/CVPR.2018.00116), URL: http://openaccess.thecvf.com/content/_cvpr/_2018/html/Zhou_Learning_Rich_Features_CVPR_2018_paper.html.
- [417] Wangchunshu Zhou and Ke Xu, « Learning to Compare for Better Training and Evaluation of Open Domain Natural Language Generation Models », in: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, AAAI Press, 2020, pp. 9717–9724, URL: <https://ojs.aaai.org/index.php/AAAI/article/view/6521>.
- [418] Wangchunshu Zhou et al., « Self-Adversarial Learning with Comparative Discrimination for Text Generation », in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020, URL: <https://openreview.net/forum?id=B1l8L6EtDS>.

-
- [419] Xinyi Zhou, Jindi Wu, and Reza Zafarani, « SAFE: Similarity-Aware Multi-modal Fake News Detection », in: *Advances in Knowledge Discovery and Data Mining - 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11-14, 2020, Proceedings, Part II*, ed. by Hady W. Lauw et al., vol. 12085, Lecture Notes in Computer Science, Springer, 2020, pp. 354–367, DOI: [10.1007/978-3-030-47436-2_27](https://doi.org/10.1007/978-3-030-47436-2_27), URL: https://doi.org/10.1007/978-3-030-47436-2_27.
- [420] Xinyi Zhou and Reza Zafarani, « Fake News: A Survey of Research, Detection Methods, and Opportunities », in: *CoRR* abs/1812.00315 (2018), arXiv: [1812.00315](https://arxiv.org/abs/1812.00315), URL: <http://arxiv.org/abs/1812.00315>.
- [421] Xinyi Zhou and Reza Zafarani, « Network-based Fake News Detection: A Pattern-driven Approach », in: *SIGKDD Explor.* 21.2 (2019), pp. 48–60, DOI: [10.1145/3373464.3373473](https://doi.org/10.1145/3373464.3373473), URL: <https://doi.org/10.1145/3373464.3373473>.
- [422] Xinyi Zhou et al., « Fake News Early Detection: A Theory-Driven Model », in: *Digital Threats* 1.2 (June 2020), ISSN: 2692-1626, DOI: [10.1145/3377478](https://doi.org/10.1145/3377478), URL: <https://doi.org/10.1145/3377478>.
- [423] Xinyi Zhou et al., « Fake News: Fundamental Theories, Detection Strategies and Challenges », in: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM '19, Melbourne VIC, Australia: Association for Computing Machinery, 2019*, 836–837, ISBN: 9781450359405, DOI: [10.1145/3289600.3291382](https://doi.org/10.1145/3289600.3291382), URL: <https://doi.org/10.1145/3289600.3291382>.
- [424] Xinyi Zhou et al., « "This is Fake! Shared it by Mistake": Assessing the Intent of Fake News Spreaders », in: *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, ed. by Frédérique Laforest et al., ACM, 2022, pp. 3685–3694, DOI: [10.1145/3485447.3512264](https://doi.org/10.1145/3485447.3512264), URL: <https://doi.org/10.1145/3485447.3512264>.
- [425] Jiren Zhu et al., *HiDDeN: Hiding Data With Deep Networks*, 2018, arXiv: [1807.09937](https://arxiv.org/abs/1807.09937) [cs.CV].
- [426] Yaoming Zhu et al., « Taxygen: A Benchmarking Platform for Text Generation Models », in: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, ed. by Kevyn Collins-Thompson et al., ACM, 2018, pp. 1097–1100, DOI: [10.1145/3209978.3210080](https://doi.org/10.1145/3209978.3210080), URL: <https://doi.org/10.1145/3209978.3210080>.

-
- [427] Brian D Ziebart, « Modeling purposeful adaptive behavior with the principle of maximum causal entropy », PhD thesis, figshare, 2010.
- [428] George Kingsley Zipf, *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*, Addison-Wesley Press, 1949, URL: <https://books.google.fr/books?id=1tx9AAAAIAAJ>.
- [429] Andy Zou et al., *Universal and Transferable Adversarial Attacks on Aligned Language Models*, 2023, arXiv: [2307.15043](https://arxiv.org/abs/2307.15043) [cs.CL].
- [430] Arkaitz Zubiaga, Maria Liakata, and Rob Procter, « Exploiting Context for Rumour Detection in Social Media », in: *Social Informatics - 9th International Conference, SocInfo 2017, Oxford, UK, September 13-15, 2017, Proceedings, Part I*, ed. by Giovanni Luca Ciampaglia, Afra J. Mashhadi, and Taha Yasseri, vol. 10539, Lecture Notes in Computer Science, Springer, 2017, pp. 109–123, DOI: [10.1007/978-3-319-67217-5_8](https://doi.org/10.1007/978-3-319-67217-5_8), URL: https://doi.org/10.1007/978-3-319-67217-5_8.

LIST OF FIGURES

1	Example of image repurposing. The original picture has been taken after the 420 festival, yet the author claims that it has been taken after an environmental demonstration in order to discredit the movement.	4
2	Representation of an embedding space adapted to cats and dogs picture classification.	8
3	Example of an original article from Reuters (A) and one generated following our methodology with Grover [405] (B), headlines are in bold.	13
1.1	Visualisation of the attention mechanism for the word “he” in one attention layer of a Transformer. The weights are computed using Equation 1.1, the higher the weight (darker color), the more the token influences the representation of the word in the next layer.	22
1.2	Illustration of the original Transformer encoder-decoder architecture used for a translation task (a) and the different components of the blocks composing the two stacks (b). Illustration taken from the Illustrated Transformer . . .	24
1.3	Representation of the different attention masks used in the different architecture of Transformers, from left-to-right: encoder-only, encoder-decoder and decoder only. Black cells represented masked tokens, that is, tokens not attended to compute the representation of the token.	25
2.1	Different training scenarios: usual text classification framework (here with a standard BERT classifier), generated data used as substitution (especially useful when the original data cannot be shared), generated data as complement.	35
2.2	Examples of tweets artificially generated with a GPT-2 model trained on the MediaEval examples with class \mathcal{T}_{5G}	38
2.3	Examples of artificial reviews generated with a GPT-2 model trained on the CLS-FR examples with the class $\mathcal{T}_{negatif}$	39

2.4	Confusion matrix of the FlauBERT / \mathcal{T} (red) and FlauBERT / \mathcal{G}^f (green) models on the CLS-FR data. The Venn diagrams shows the proportions of shared examples for each category.	43
2.5	Performance (macro-F1) according to the quality (accuracy in %) of the classifier filtering the artificially generated data; MediaEval dataset with logistic regression.	45
4.1	Illustration of generative discriminators. Class-conditional language models compute the probability of the next token conditioned on a class using a control code. The classification probability is then computed using Bayes' rule.	56
5.1	Illustration of the constrained generation process as a tree exploration from the prompt The cat . Classification probabilities are only represented on completed sequences.	61
5.2	MCTS application to text generation.	63
5.3	Example of two constrained generations using PPL-MCTS, one on the negative class, one on the positive class, using the same prompt (in bold) from amazon_polarity and same language model.	67
5.4	Examples of constrained generation using PPL-MCTS, PPLM, GeDi and Sampling - Argmax methods (from top to bottom) on the positive class of amazon_polarity, using the same prompt (in bold).	69
5.5	Examples of constrained generation using PPL-MCTS, PPLM, GeDi and Sampling - Argmax methods (from top to bottom) on the 'love' class from 'emotion', using the same prompt (in bold).	70
5.6	Accuracy (left) and perplexity (right) of the Sampling - Argmax method according to the α parameter; amazon_polarity dataset	71
5.7	Accuracy according to the roll-out size; CLS dataset	71

6.1	Training GANs vs GCNs. Solid arrows stand for sampling, dashed arrows depict dependence between distributions and dotted ones denote training. (a) Classical discrete GAN where: 1) the generator p samples sequences; 2) the discriminator D is updated according to these generated sequences and those from the training set Γ ; 3) the scores given by the discriminator D are used as the reward in a policy gradient update of p . (b) Our GCN approach where \hat{q} and q respectively stand as the behavior and target distributions, which are defined by a cooperative scheme between D and p . After sampling from \hat{q} in 3), q is used in an importance sampling weight for the update of p in 4), which corresponds to the minimization of $KL(q p)$. The distribution q is set as $q(x) \propto p(x)D(x)$ to ensure convergence, and \hat{q} can take various forms, the closer to q the lower the variance.	83
6.2	Examples of questions generated using GCN. The questions to be generated are conditioned on the input context and the expected answer.	87
6.3	Results on the EMNLP 2017 dataset. Left: Evolution of BLEU results on tests sets w.r.t. training epochs (higher is better) – red: GAN without scheduler, blue: GAN with scheduler, green: GCN. Right: Curves of negative BLEU vs self BLEU (lower is better). Scores for previous studies are taken from [224].	89
6.4	Evolution of performance on the test set w.r.t. training epochs (in term of BLEU, the higher the better), for conditioned NLG tasks. Left: Question Generation, Right: Summarization.	90
7.1	Illustration of the Therapy method. Texts from different classes are cooperatively generated using the guidance of the studied model. A logistic regression is then trained to predict the label of the generated texts. The weights of the model associated with each word are then returned as importance weights.	98
7.2	Spearman correlation w.r.t number of generated text per class for amazon_polarity and ag_news.	99
7.3	Samples generated by Therapy and top words returned by the method for both classes of the amazon_polarity dataset.	101
7.4	Precision/recall curves of the glass-box top words for the different explanation methods.	104

7.5	Proportion of texts whose glass-box prediction changes w.r.t the number of important words from the original class replaced by important words from other classes.	104
8.1	Accuracy (%) of the different type of discriminators w.r.t. the input length (# tokens)	112
8.2	Execution time of MCTS iterations (s) w.r.t. generation step (averaged over 10 batchs of 30 sequences)	115
9.1	Representations of the attention layers of different architectures for multi-modal Transformers: (a) single stream, (b) dual stream, and (c) dual encoder. Note that dual stream Transformers also use monomodal self-attention layers interleaved with cross-modal attention layers that are not represented for the sake of simplicity.	120
9.2	Representation of a joint (left) and a common (right) embedding space. An image and a corresponding text are projected together in the joint embedding space, where the couple can be compared to another couple, whereas modalities are projected independently in the common space and can be compared to either the same modality or the other modality.	122
9.3	Illustration of the cross-modal contrastive objective, taken from the CLIP paper [270]. The model is trained to maximize the diagonal and minimize other elements in rows/columns.	125
10.1	Examples of images with an overly generic ground truth caption, a caption generated by a model without regularization (leading to reward hacking), and the caption generated by our approach (well written and distinctive).	129
10.2	Proposed captioning model learning overview. Generated and ground-truth captions, as well as input and mined similar images, are projected in the CLIP embedding space. Those representations are used to compute the reward composed of a discriminator score (Section 10.3.1) and a CLIP-based bidirectional contrastive similarity score (Section 10.3.2), for <i>beam search</i> and <i>ground-truth</i> samples (Section 10.3.3) (in green in the reward computation bloc).	132

LIST OF TABLES

2.1	Performance (%) of neural classification approaches on the MediaEval, CLS-FR, and AG_news tasks according to the scenarios of usage of the artificially generated texts without filtering. The BERT* model are respectively RoBERTa, FlauBERT and RoBERTa.	41
2.2	Performance (%) of neural classification approaches on the MediaEval, CLS-FR, and AG_news tasks according to the scenarios of usage of the artificially generated texts after filtering. The BERT* model are respectively RoBERTa, FlauBERT and RoBERTa.	42
2.3	Performance (%) of the LR/bag-of-words approach on the MediaEval, CLS-FR and AG_news datasets according to our scenarios of usage of the artificially generated data after filtering: without, substitution, complement. .	44
5.1	Performance of constrained generation methods; from left to right: amazon_polarity, emotion, CLS datasets. † (resp. *) indicates statistically significant improvement against GeDi-classloss (resp. PPLM).	68
5.2	Results of the human evaluation on the CLS dataset (averaged over 3 annotators). * indicates statistically significant ($p \leq 1\%$) improvement against PPLM.	73
6.1	Final results on QG and Summarization test sets, in terms of BLEU-4 (B), ROUGE-1 (R-1) and ROUGE-L (R-L). Scores in bold are significantly different from the best baseline ($\text{GAN}_{+scheduler}^{\hat{q}=MCTS}$) according to a 95%-Student-t-test.	91

7.1	Spearman correlation (p-value) between the top words of a logistic regression glass-box and explanation methods learning a logistic regression over generated texts. Baseline uses unconstrained samples while Therapy generates samples using the MCTS, either selecting the most played or highest scored node. Results are shown per class and dataset.	102
7.2	Spearman correlation (p-value) between the top words of a logistic regression glass-box and the four explanation methods. ‘other’ indicates that the explanations are generated using the other dataset. Results are shown per class and dataset.	102
8.1	Performance of MCTS w.r.t. the metric to optimize on amazon_polarity (left) and AG_news (right) datasets. * indicates statistically significant improvement against Generative Discriminator. Note that no model demonstrated significant improvement over the unidirectional discriminator. . . .	114
10.1	Captioning results on the MS COCO dataset (Karpathy splits). R@k correspond to the retrieval rate at k using the fixed CLIP model either using text queries (T2I) or image queries (I2T). Writing quality metrics includes BLEU@4 (B4), ROUGE-L (R-L), CIDEr (C), METEOR (M) and SPICE (S). The Self-BLEU metric measures the diversity.	138
A.1	Results for every tested set of parameters on the proposed methods; emotion dataset. Results reported in the main body are in italic.	156
A.2	Results for every tested set of parameters on the proposed methods; CLS dataset. Results reported in the main body are in italic.	157
A.3	Results for every tested set of parameters on the proposed methods; amazon_polarity dataset. Results reported in the main body are in italic. . . .	158
B.1	Final results on QG and Summarization test sets, in terms of BLEU-4 (B), ROUGE-1 (R-1) and ROUGE-L (R-L), using the same base model as used in [305].	163



Titre : Détection de désinformation multimodale : surmonter le défi de la collecte de données d'entraînement grâce à la génération de données

Mot clés : détection de désinformation, données d'entraînement, modèles génératifs, génération coopérative, réseaux antagonistes génératifs, multimodalité

Résumé : Pour répondre au problème croissant de la désinformation, des outils de vérification automatique de l'information sont nécessaires. Des images étant fréquemment associées à la désinformation, ces modèles doivent être multimodaux. La collecte de suffisamment de données non biaisées nécessaires pour entraîner les modèles est un défi. Dans cette thèse, nous explorons comment les modèles génératifs peuvent être utilisés pour des tâches discriminatives en cas de manque de données. Pour résoudre le problème des récompenses clairsemées des GAN textuels, nous explorons la génération coopérative, où le générateur est guidé par un modèle externe, et nous présentons une

méthode originale basée sur le MCTS. Ensuite, nous utilisons la génération coopérative pour créer des explications de modèles boîte noire et réalisons une étude empirique sur la complexité/qualité de différents types de modèles dans le cadre de cette coopération. Enfin, nous explorons l'utilisation de légendes humaines dans l'apprentissage par renforcement d'un modèle de légendage d'images en utilisant des récompenses d'un modèle de recherche cross-modal. Nous concluons en discutant des opportunités et des risques des modèles génératifs dans le contexte de la désinformation et en abordant la question du tatouage numérique.

Title: Multimodal misinformation detection: overcoming the training data collection challenge through data generation

Keywords: misinformation detection, training data, generative models, cooperative generation, generative adversarial networks, multimodality

Abstract: To tackle the growing issue of misinformation, automated fact-check tools are required. Because images are often found within misinformation, these models need to be multimodal. Collecting enough unbiased data to train the models is challenging. In this thesis, we explore how generative models can be used for discriminative tasks when there is a lack of data. To tackle the sparse rewards issue of textual GANs, we explore cooperative generation where the generator is guided by an external model and present a novel method

based on the MCTS. We then use cooperative generation to generate explanations of black-box models and conduct an empirical study on the complexity/quality of different types of models in the cooperative setup. Finally, we explore the use of ground truth caption in a reinforcement learning training of an image captioning model using rewards from a cross-modal retriever. We conclude by discussing the opportunities and risks of generative models in the context of misinformation as well as watermarking.