Encodeur hybride pour la détection automatique de désinformation

Géraud Faye*, Sylvain Gatepaille*, Guillaume Gadek*, Souhir Gahbiche*

* Airbus, Élancourt, France
{geraud.faye, sylvain.gatepaille, guillaume.gadek, souhir.gahbiche}@airbus.com

Résumé. L'encodage de texte pour des tâches de classification repose aujourd'hui grandement sur de larges modèles de langage difficilement explicables et nécessitant de grandes quantités de données pour fonctionner. Ces modèles sont à la base de tâches de classification comme la détection de désinformation, importante aujourd'hui. Récemment, les approches hybrides entre l'apprentissage profond et l'IA symbolique tentent de surpasser les performances des modèles à base d'attention en introduisant du raisonnement dans le processus de décision pour le rendre moins opaque à l'utilisateur. Dans cet article, nous proposons CATS, un mécanisme d'attention basé sur la compréhension sémantique des documents, améliorant les performances des modèles neuronaux équivalents, réduisant le besoin en données annotées et facilitant l'explicabilité de la décision.

1 Introduction

Avec le récent et fort développement des réseaux sociaux, la diffusion de désinformation est devenue de plus en plus présente, au point où une majorité la considère comme une menace pour la démocratie [1]. Les réseaux sociaux deviennent l'unique source d'information pour de plus en plus de personnes [2], ce qui en fait le terrain idéal pour la désinformation. Il s'agit d'une certaine forme de mésinformation (information de mauvaise qualité) s'appuyant sur des biais cognitifs (Greifeneder et al., 2020) afin d'influer sur l'opinion publique.

Étant par nature liée à l'actualité et à la politique, la désinformation peut avoir un impact important lorsqu'elle est utilisée pour manipuler les votes lors d'élections majeures (élections américaines de 2016 ou référendum du Brexit) ou pour impacter la santé publique (récente crise du Covid-19 ou les vaccins en général). La manière dont ces informations sont rédigées est idéale pour les réseaux sociaux, car elle fait réagir et incite au partage. De plus, il est très difficile de réparer les dommages causés une fois qu'elles ont été beaucoup lues (loi de Brandolini). Cela rend important leur détection avant qu'elles ne soient largement partagées. La grande quantité d'information postée quotidiennement rend l'automatisation de cette tâche de détection cruciale.

Les travaux actuels se concentrent soit sur des facteurs de style des articles (utilisation de *features* de modèles transformers, règles symboliques), soit sur des facteurs de propagation

1. www.civica.eu/fake-news-and-democracy/
2. en.wikipedia.org/wiki/Social_media_as_a_news_source

sur internet, car les schémas de propagation de la désinformation sont différents de ceux des informations légitimes. D'autres approches utilisent les premiers commentaires des articles ou le contenu de tweets les citant afin de mieux détecter la désinformation.

Toutefois, comme nous nous intéressons à la détection de la désinformation en amont, nous faisons le choix de traiter la désinformation uniquement sous forme textuelle. Nous proposons une adaptation des modèles d'attention avec un nouveau mécanisme d'attention à base de règles **CATS** (***C**ognitive **A**ttention **T**o **S**emantics*) qui essaie de reproduire l'attention cognitive humaine pour la compréhension des textes.

En section 2, nous allons présenter l'état de l'art de la détection de fake news et l'interprétation actuelle des modèles d'attention. En section 3, nous proposons une nouvelle approche de l'attention basée sur la sémantique, qui sera ensuite évaluée en section 4 face à des modèles plus classiques.

2 État de l'art

2.1 Considérations autour de la désinformation

La désinformation/mésinformation est relativement complexe à définir et sa définition varie souvent en fonction des auteurs ou des jeux de données. Islam et al. (2020) ont identifié 5 grandes catégories non-exclusives de mésinformation :

— Fausse information : le contenu factuel de l'information est faux et peut être discrédité.
— Rumeur : le contenu de l'information ne peut être vérifié de manière certaine.
— Spam : l'information est propagée de manière répétée pour induire de la confusion chez les lecteurs.
— *Fake news* : l'information, bien que basée sur des faits réels, est modifiée pour ne plus correspondre à la réalité, tout en restant plausible.
— Désinformation : l'information est reportée avec une intention de tromper le lecteur.

Ces différentes catégories sont souvent réunies dans les jeux de données sous le même label, les informations rapportées étant dans tous les cas inexactes. Différentes perspectives permettent de détecter cette désinformation, par lesquelles elle diffère de l'information légitime (Zhou et Zafarani, 2018) :

— Identification des faits relayés : ces derniers sont généralement faux dans les articles de désinformation.
— Style d'écriture : afin d'être plus relayée par les utilisateurs, l'information est écrite pour appeler à l'émotion et à la réaction, souvent dans un style journalistique pauvre.
— Schémas de propagation : la manière avec laquelle la désinformation est relayée diffère grandement de celle des informations légitimes, la désinformation se propageant près de 6 fois plus vite en moyenne.

Les modèles de détection automatique de désinformation font le choix de se focaliser sur un seul de ces points, ou alors les combinent pour obtenir une prédiction plus fiable.

2.2 Méthodes de détection de la désinformation

Une fois propagée et largement commentée, la désinformation est facile à identifier. En utilisant les schémas de propagation (graphe reliant l'information, les commentaires la men-

tionnant et les commentaires supplémentaires), Han et al. (2020) parviennent à identifier la désinformation sans accéder à son contenu. En y ajoutant un graphe de posture des utilisateurs, Davoudi et al. (2022) parviennent à identifier de manière presque parfaite la désinformation. En ajoutant du contenu textuel, Lu et Li (2020) parviennent à identifier les tweets propageant de la désinformation sans citer d'articles de presse.

Toutefois, ces méthodes utilisent les schémas de propagation de l'information, qui ne devraient pas être disponibles si la désinformation était identifiée en amont, ce qui est notre problème d'intérêt.

Les modèles de langages pré-entrainés sont étonnamment performants une fois *fine-tunés* pour identifier la désinformation (Pelrine et al., 2021). Certaines méthodes sont plus complexes, utilisant des bases de données d'articles vrais (Vo et Lee, 2021), des architectures plus profondes (Karnyoto et al., 2021) ou des réseaux adversariaux (Wang et al., 2018).

Certains auteurs montrent que les différents types de mésinformation partagent des points communs en entrainant un encodeur commun sur différentes tâches (rumeurs, pièges à clics, désinformation...), ce qui a eu pour conséquence d'améliorer la performance sur chacune des tâches considérées (Lee et al., 2021). D'autres modèles plus simples existent, comme celui présenté par Guélorget et al. (2021), utilisant un réseau convolutif pour obtenir des résultats satisfaisants et interprétables.

Pour limiter la sur-spécialisation sur certains sujets, Castelo et al. (2019) proposent de conjuguer l'usage d'un modèle profond faisant la classification avec un discriminateur se chargeant d'identifier l'évènement traité. L'objectif est de pénaliser le réseau si les *embeddings* produits sont spécifiques à différents évènements, afin d'obtenir une meilleure généralisation.

Il existe de nombreux jeux de données, avec des articles souvent annotés manuellement par des associations de journalistes telles que Politifact [3]. Les principaux datasets contenant des informations de propagation sont consignés dans le tableau 1.

Nom	Sujets
PHEME (Lukasik et al., 2015)	9 évènements divers
WNUT-2020 (Nguyen et al., 2020)	Covid-19
PolitiFact (Shu et al., 2018)	Politique
GossipCop (Shu et al., 2018)	Actualités *people*

TAB. 1 – Principaux datasets de désinformation.

2.3 Fonctionnement des modèles d'attention

Afin de comprendre l'intuition qui a mené à notre modèle, il faut rappeler le fonctionnement du mécanisme d'attention des transformers (Vaswani et al., 2017). Les *embeddings* en entrée d'une couche d'attention sont projetés de manière linéaire dans 3 espaces nommés *Key*, *Query* et *Value*, donnant trois matrices K, Q et V. On calcule ensuite la matrice d'attention A en multipliant les matrices K et Q entre elles, puis on applique une normalisation *softmax* par ligne, donnant $A = \text{softmax}(QK^T)$

3. www.politifact.com/

Cette matrice est ensuite multipliée par la matrice V pour obtenir la sortie de la couche d'attention. Chaque ligne de la matrice d'attention A décrit comment les *embeddings* de chaque token influent sur le token correspondant à la ligne. Ce mécanisme ressemble au principe de compositionnalité sémantique, une théorie de la compréhension sémantique du langage [4].

Ce mécanisme continue de faire l'objet de travaux et certains essaient de modifier son fonctionnement tout en gardant la même inspiration. Par exemple, les *synthesizers* (Tay et al., 2020) calculent l'attention avec un réseau de neurones entièrement connecté, ou même des poids aléatoires, tout en gardant des performances similaires aux *transformers* (Vaswani et al., 2017).

Des travaux plus spécifiques sur la compréhension des modèles d'attention existent. Vig (2019) a développé un outil permettant de visualiser les poids d'attention dans un transformer sur des exemples. Cet outil a permis le développement de la BERTologie (Rogers et al., 2020) qui vise à la compréhension des mécanismes d'attention. Pande et al. (2021) ont identifié différents comportements syntactiques proches des raisonnements humains (liaison entre verbe et sujet, entre nom et adjectif, ...). Cette interprétation des têtes d'attention laisse entrevoir la possibilité d'utiliser des règles sémantiques explicites permettant de relier les mots directement dans la matrice d'attention, sans avoir à passer par les matrices K et Q qui nécessitent d'être apprises.

Le fonctionnement observé des têtes d'attention nous ouvre la possibilité de reproduire son fonctionnement, mais en se basant directement sur l'attention cognitive humaine que les transformers essaient d'imiter. Cette attention cognitive basée directement sur des règles compréhensibles pourrait rendre les systèmes de traitement du langage plus robustes et interprétables.

3 Modèles proposés

3.1 CATS - Attention Cognitive Sémantique

La principale contribution de ce papier est l'introduction du mécanisme d'attention cognitive sémantique CATS (**C**ognitive **A**ttention **T**o **S**emantics), basé sur les interprétations de l'attention par la BERTologie.

Pour l'esprit humain, le sens des mots dépend de leur contexte, et le sens de chaque mot est altéré par leur rôle dans la phrase. Par exemple, un adjectif vient moduler le sens d'un nom ou alors un sujet va moduler le sens du verbe auquel il est attaché. La BERTologie s'est aperçue qu'une partie des têtes d'attention effectuait cette même modulation, ce qui a inspiré cette couche réutilisable dans tous les modèles traitant des données textuelles.

Dans un premier temps, la couche reçoit en plus des *embeddings* de chacun des *tokens* la phrase entière sous forme de chaîne de caractères afin de pouvoir la traiter sémantiquement. L'analyse sémantique est effectuée avec SpaCy [5]. Cet outil permet à partir d'une phrase de créer un arbre sémantique reliant chaque mot avec les autres en fonction de leur rôle dans la phrase. Un exemple d'arbre sémantique est donné en figure 1.

Les adjectifs sont directement connectés au nom correspondant et les groupes verbaux sont décomposés entre sujet, base verbale et compléments. L'entité sémantique apportant le plus de sens en général est le verbe, qui est alors à la racine de l'arbre. L'avantage de cette approche

4. https ://doi.org/10.1093/acrefore/9780199384655.013.42
5. spacy.io

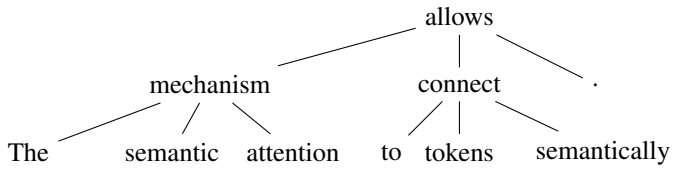

FIG. 1 – Arbre sémantique calculé par SpaCy pour la phrase “The semantic attention mechanism allows to connect tokens semantically.’.

sémantique est qu’elle peut être appliquée à tout texte et toute langue avec une analyse sémantique adaptée. Ainsi, toutes les phrases, même si elles diffèrent beaucoup de celles vues par le modèle pendant l’entraînement, pourront être encodées avec la même logique. On s’attend donc à une meilleure généralisation avec cette approche, particulièrement lorsque peu de données sont disponibles.

Une fois cet arbre construit, la matrice d’attention est remplie en fonction de ce dernier. La matrice d’attention est d’abord initialisée comme la matrice identité. L’arbre sémantique est ensuite lu du bas vers le haut. Chaque mot de l’avant-dernière couche de l’arbre est connecté avec les *tokens* au dessous en ajoutant leur ligne correspondante de la matrice d’attention multipliée par un scalaire γ dans la matrice d’attention, similaire au facteur de réduction de l’apprentissage par renforcement. Un exemple de construction de matrice est présenté en figure 2.

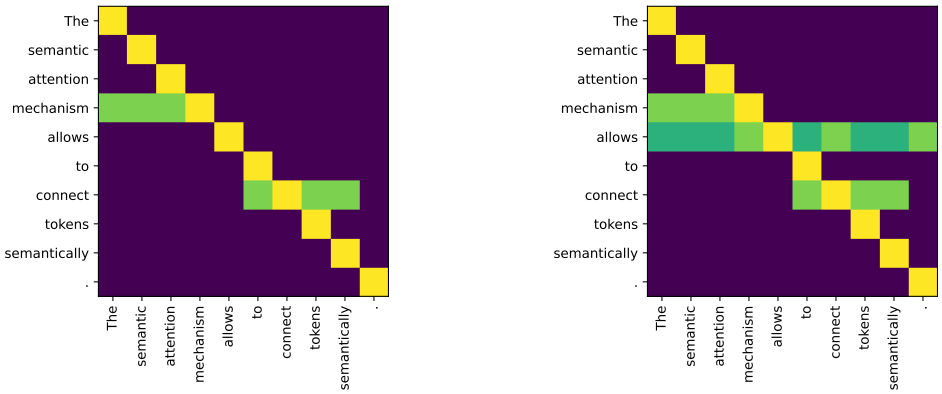

FIG. 2 – Construction de la matrice d’attention correspondant à l’arbre de la figure 1. À la première étape, seules les lignes de “mechanism” et “connect” sont mises à jour. À l’étape suivante, la ligne du mot principal “allows” est remplie à partir des lignes de “mechanism”, “connect” et “.”. Les mots connectés indirectement au *token* principal ont un poids plus faible.

Cette matrice d’attention est ensuite utilisée dans la couche d’attention classique à la place de la matrice d’attention classique A. Ce processus limite la couche d’attention à utiliser une unique tête d’attention car une seule analyse est effectuée dans notre cas. Cette couche n’a

pas de poids entraînables, ce qui économise près de 2.4 millions de poids pour une dimension d'*embeddings* de 768.

Il faut noter que, contrairement à la couche standard d'attention, les *embeddings* ne sont que *combinés* par CATS et qu'aucune projection n'est faite, même en sortie de la couche. Ce comportement fait que les couches CATS et d'attention standard ne sont pas strictement équivalentes, et que des matrices de projection sont à ajouter si l'on souhaite projeter les *embeddings* vers un autre espace latent.

3.2 Modèles à base d'attention cognitive sémantique

On définit à présent les modèles testés, qui peuvent chacun se résumer en 3 étapes :

1. Le texte est *tokenisé*, puis chaque *token* est projeté dans un espace vectoriel de dimension 300 avec fastText (Bojanowski et al., 2016), puis projeté dans un espace de dimension 768, la dimension utilisée par les *embeddings* de BERT (Devlin et al., 2018).
2. Ensuite, un mécanisme d'attention est utilisé, soit la couche standard d'attention utilisée par BERT, soit une couche CATS (respectivement notées "Standard" et "CATS" dans les résultats.
3. Enfin, une couche de classification est ajoutée. Deux options sont utilisées. Dans un premier temps, nous proposons une couche entièrement connectée (notée “dense” dans les résultats) avec une fonction d'activation softmax pour obtenir la probabilité que l'article soit de la désinformation. Toutefois, la couche CATS ne connecte pas les phrases entre elles (un arbre sémantique pour une phrase). Pour parer à cela, nous proposons un modèle récurrent (un réseau récurrent à porte “GRU”) qui traite séquentiellement les *embeddings* des *tokens* à la racine des arbres sémantiques des documents, donnant une prédiction à partir d'une analyse phrase par phrase du document.

4 Résultats et discussion

4.1 Identification de désinformation

Les modèles présentés dans la section précédente ont été testés sur les jeux de données PolitiFact et GossipCop, composés respectivement de 960 et 19545 articles de presse. Ces jeux de données ont été retenus car ils disposent aussi des métadonnées de propagation des articles, pouvant donner lieu à d'autres travaux prenant plus de paramètres en compte. Ces derniers ont été nettoyés pour garder les articles écrits (on retire les articles centrés sur des photos ou vidéos) et correctement *scrappés* par le script de FakeNewsNet [6]. Les jeux de test sont définis avant tout entraînement avec 10% des données. Le reste des données est ensuite aléatoirement séparé en jeu d'entraînement et de validation (80% et 10% des données initiales respectivement). Les résultats rapportés sont obtenus avec une moyenne et un écart-type sur 5 séparations aléatoires.

Ces modèles sont entraînés avec arrêt prématuré si le score F1 de validation n'augmente plus pendant 10 itérations. Les poids sont alors restaurés à ceux ayant eu le meilleur score F1 et le modèle est évalué sur le jeu de données de test. Les résultats sont consignés dans le tableau 2.

6. https ://github.com/KaiDMML/FakeNewsNet

Notons que pour des raisons de temps d'apprentissage, seulement 20% du dataset GossipCop ont été utilisés en tant que jeu de données d'apprentissage. À titre de comparaison, les scores obtenus par BERT *fine-tuné* avec les mêmes divisions train/validation/test sont mentionnés.

	PolitiFact		GossipCop	
	Fiabilité	F1	Fiabilité	F1
Standard + dense	0.889±0.035	0.902±0.034	0.727±0.020	0. 758±0.026
CATS + dense	**0.916**±0.029	**0.929**±0.025	0732±0.031	**0.762**±0.027
CATS + GRU	0.914±0.018	**0.929**±0.014	**0.753**±0.047	0.761±0.047
BERT	0.930±0.027	0.940±0.023	0.824±0.021	0.819±0.033

TAB. 2 – Résultats des différents modèles sur le jeu de données de test. Les meilleurs scores pour les modèles comparables sont reportés en gras.

Les résultats globaux démontrent la capacité d'expression de CATS pour l'encodage de textes en comparaison avec son équivalent entièrement neuronal et en comparaison à BERT pour PolitiFact. On remarque une différence notable en la capacité de discrimination entre les deux datasets. Cela vient du fait que les sujets traités dans les articles de PolitiFact ont tendance à donner des articles de désinformation dans un style plus agressif et très différent du style des informations réelles. Dans le cadre de GossipCop et de ses articles concernant des personnalités, la différence de style entre un article propageant une rumeur et un article propageant une vraie information est moindre, ce qui explique aussi la chute de score pour le mécanisme d'attention standard et BERT.

4.2 Avantages par rapport à un modèle entièrement neuronal

Les modèles utilisant CATS sont **neuro-symboliques** dans la mesure où ils manipulent les données sous forme de vecteurs scalaires (*embeddings*) et symbolique (graphe sémantique). Ce type de modèles est théorisé par certains auteurs (Kautz, 2022) comme le futur de l'intelligence artificielle et pourrait apporter des avantages par rapport aux modèles profonds :

— Modèles plus légers en termes de poids
— Besoin réduit en données annotées
— Meilleure interprétabilité des modèles

Nous allons dans les prochaines sous-sections étudier si notre nouvelle approche présente ces avantages.

4.2.1 Analyse du nombre de paramètres et des temps de calcul

Le passage d'un type de modèle entièrement neuronal à un type de modèle neuro-symbolique a un coût important en temps de calcul, qui a été mesuré. Les résultats en coût temporel et en gain de poids du modèle sont présentés dans le tableau 3. La colonne *Forward* correspond au temps de calcul effectif une fois que les étapes de pré-calcul ont été réalisées.

L'analyse sémantique du texte est la partie la plus longue du modèle, le rendant plus de 30 fois plus long pour le traitement de nouvelles données. Cette analyse peut cependant se faire une unique fois par jeu de données, ne rendant finalement le modèle que 2,5 fois plus lent une

	Pré-calcul	*Forward*	#paramètres
Standard + dense	**0s**	**17ms**	3.0M
CATS + dense	0.52s	43ms	**0.6M**
CATS + GRU	0.52s	111ms	29.3M

TAB. 3 – Ressources temporelles et spatiales demandées par les différents modèles. Les valeurs en gras sont celles des modèles les plus efficaces.

fois le pré-calcul terminé (43ms contre 17ms). Cette optimisation du calcul est primordiale pour rendre l'entraînement possible malgré l'ajout coûteux de l'analyseur sémantique.

Cette augmentation de temps de calcul est encore plus grande pour le modèle utilisant un réseau récurrent (111ms contre 17ms), car ce dernier rajoute plus de 25 millions de paramètres. Cependant, le gain en nombre de paramètres est substantiel pour les modèles plus simples, avec la disparition des matrices projetant les *embeddings* dans les espaces *Key*, *Query* et *Value*.

De plus, le temps de pré-calcul n'a lieu qu'une unique fois, et la matrice d'attention peut être partagée entre différentes couches, comme dans l'extension AlBERT de BERT qui partage les paramètres entre les différentes couches des transformers, rendant cette approche *scalable*.

4.2.2 Besoin en données annotées

Pour tester le besoin en données annotées, nous avons réentraîné chaque modèle sur des fractions de PolitiFact. Nous avons appliqué la même méthodologie qu'énoncée au début de la section. La dépendance en données annotées est illustrée dans la figure 3.

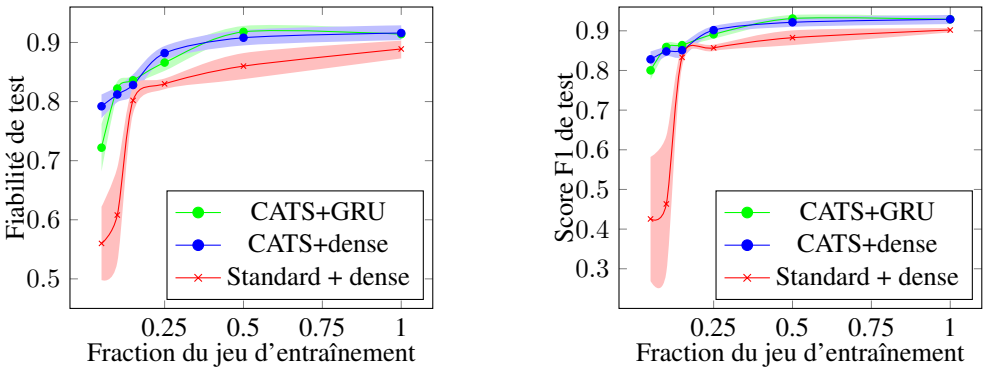

FIG. 3 – Évolution des différentes métriques lors d'un apprentissage sur une portion du jeu de données PolitiFact.

Les résultats obtenus montrent que notre modèle avec une couche de classification dense est bien efficace lorsque les données annotées sont rares, avec un gain de près de 40% face à son équivalent neuronal lorsque l'on utilise uniquement 5% du jeu d'entraînement (25 articles). Le modèle avec GRU a un avantage moindre, dû au réseau récurrent qui nécessite tout de même beaucoup de données.

Ce comportement valide notre supposition que le modèle ait besoin de moins de données annotées. L'utilisation du moteur de raisonnement symbolique permet d'encoder efficacement les phrases du jeu de test, quelle que soit leur structure. On obtient facilement des *embeddings* assez discriminants pour la classification de texte, sans avoir eu à entraîner un grand modèle de langage comme BERT. Cette propriété est utile pour résoudre les tâches avec peu de données.

4.2.3 Interprétabilité

Les grands modèles de langage ont le désavantage d'être très peu explicables, limitant leur utilisation dans certains domaines critiques, comme la modération de contenu fiable. Toutefois, les modèles d'attention avec peu de couches sont interprétables car on peut calculer la contribution de chaque token dans la décision finale à partir des poids d'attention et des embeddings. Cela est possible car la matrice d'attention produite par CATS est inversible et que l'on peut aussi *inverser* la couche de classification.

La couche de classification aplanit la matrice d'embeddings puis applique une opération linéaire avant d'utiliser la fonction softmax. Les contributions de chacun des *tokens* peuvent être isolées, donnant un vecteur de contributions Z avec les coordonnées vérifiant $Z_i = WX_i + b$ avec W et b les paramètres de la couche de classification et X_i la matrice d'embeddings aplanie avec les embeddings des tokens $j \neq i$ masqués.

La construction d'une matrice d'attention par CATS pour une unique phrase revient à effectuer un pivot de Gauss. En effet, en partant du bas de l'arbre, on ajoute sur les lignes des noeuds supérieurs les lignes de la matrice des noeuds directement connectés en dessous, multipliées par le facteur de réduction. En effectuant ce procédé dans l'autre sens, on peut aisément calculer l'inverse de cette matrice. Quand il y a plusieurs phrases et donc plusieurs arbres, on a une matrice d'attention définie par blocs, dont chaque bloc diagonal est inversible, ce qui rend la matrice d'attention inversible.

Pour obtenir les contributions réelles C de chaque token, on multiplie le vecteur des contributions Z calculé précédemment et on le multiplie par l'inverse de la matrice d'attention, selon l'équation 1.

$$C = A^{-1}Z = A^{-1}\left(WX_i + b\right)_i \tag{1}$$

Nous avons à partir de ces calculs développé un démonstrateur permettant de visualiser ces contributions ainsi que les prédictions du modèle. Un exemple est montré en figure 4. Les poids d'attention montrés correspondent à la moyenne des poids visant le token correspondant, c'est à dire la moyenne de la colonne correspondante dans la matrice d'attention.

Dans notre exemple, il est difficile d'interpréter les poids d'attention, étant focalisés sur le premier token, de manière semblable à ce qui est observé dans BERT avec une focalisation vers le token [CLS]. Cela n'aide toutefois pas à comprendre quels tokens ont le plus guidé la décision.

Grâce à notre approche, on voit dans notre exemple que c'est la partie de l'article citant la source qui a le plus aidé à identifier de l'information légitime ("says chief warden" : *a dit le gardien en chef* sont les tokens les plus importants pour la décision). Il faut toutefois noter que le modèle ne donne pas de garantie que la source soit fiable, mais souligne le fait que l'article cite ses sources, ce qui traduit une meilleure probabilité d'information légitime. Notre

FIG. 4 – Notre démonstrateur basé sur le modèle présenté dans cet article avec un exemple d'article légitime. La visualisation des contributions de chaque token (Word contributions) apporte plus d'information que la visualisation des poids d'attention (Attention weights).

mécanisme d'attention CATS permet d'obtenir plus facilement les contributions de chacun des tokens, grâce à sa matrice d'attention inversible et basée sur des règles sémantiques.

4.3 Futures extensions

La plus grande limitation de notre modèle vient de la forte augmentation des temps de calcul par rapport à un modèle uniquement neuronal. Une première optimisation serait l'utilisation de matrices creuses pour les poids d'attention, étant donné que la plupart des poids d'attention valent strictement 0. Cette optimisation pourrait considérablement réduire la taille de la matrice d'attention en mémoire par au moins un facteur 100 sur le dataset PolitiFact.

Un autre problème inhérent à l'analyse sémantique est que chaque phrase est cloisonnée des autres dans la matrice d'attention. L'ajout d'un mécanisme de coréférence entre les phrases pourrait permettre de "fusionner" les arbres entre eux au niveau des tokens représentant les mêmes objets, ce qui enrichirait encore plus leur représentation (voir figure 5 pour un exemple).

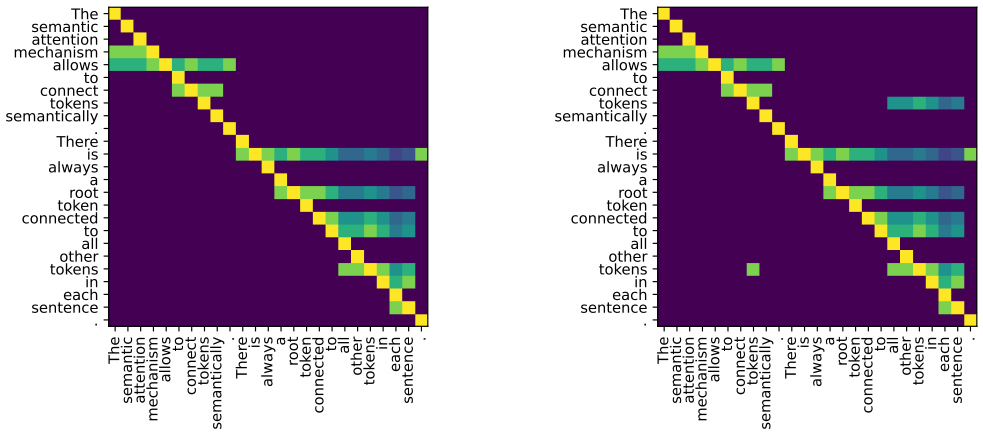

FIG. 5 – Deux matrices d'attention. À gauche, la matrice calculée par le modèle proposé et à droite, la matrice qui serait calculée avec l'extension du modèle par la coréférence. Notons que "root token" ne représente pas la même entité que les tokens indépendants dans la phrase, sa représentation n'est donc pas affectée par les autres occurences des mots *token* ou *tokens*.

Enfin, les performances obtenues sur GossipCop montrent que le fonctionnement de l'attention n'est pas parfaitement reproduit et que des études complémentaires sur le fonctionnement de BERT sur ce dataset permettrait d'identifier de nouvelles règles à utiliser dans CATS.

5 Conclusion

Nous avons dans cet article présenté la première approche hybride du mécanisme d'attention à notre connaissance. Son fonctionnement basé explicitement sur la compréhension cognitive humaine des textes est inversible, permet de facilement expliquer la prise de décision des modèles l'utilisant et fonctionne particulièrement bien lorsque peu de données sont disponibles.

Les scores obtenus pour la détection de désinformation sont encourageants, particulièrement lorsque les données annotées sont en quantité réduite. Nous pensons qu'en utilisant un mécanisme de partage des poids d'attention, cette approche pourrait être *scalable* à de grands modèles de langage. Toutefois, des optimisations et améliorations sont possibles, notamment en optimisation de temps de calcul, afin de rendre le modèle plus attractif lorsque les données annotées sont disponibles en grandes quantités.

Références

Bojanowski, P., E. Grave, A. Joulin, et T. Mikolov (2016). Enriching word vectors with subword information. *CoRR abs/1607.04606*.

Castelo, S., T. G. Almeida, A. Elghafari, A. S. R. Santos, K. Pham, E. F. Nakamura, et J. Freire (2019). A topic-agnostic approach for identifying fake news pages. *CoRR abs/1905.00957*.

Davoudi, M., M. Moosavi, et M. Sadreddini (2022). Dss : A hybrid deep model for fake news detection using propagation tree and stance network. *Expert Systems with Applications*.

Devlin, J., M. Chang, K. Lee, et K. Toutanova (2018). BERT : pre-training of deep bidirectional transformers for language understanding. *CoRR abs/1810.04805*.

Greifeneder, R., Jaffé, M.E., Newman, E.J., Schwarz, et N. (Eds.) (2020). The psychology of fake news: Accepting, sharing, and correcting misinformation (1st ed.). *Routledge*.

Guélorget, P., B. Icard, G. Gadek, S. Gahbiche, S. Gatepaille, G. Atemezing, et P. Égré (2021). Combining vagueness detection with deep learning to identify fake news. *CoRR abs/2110.14780*.

Han, Y., S. Karunasekera, et C. Leckie (2020). Graph neural networks with continual learning for fake news detection from social media. *CoRR abs/2007.03316*.

Islam, M. R., S. Liu, et G. Wang, Xu (2020). Deep learning for misinformation detection on online social networks: a survey and new perspectives. *Social Network Analysis and Mining*.

Karnyoto, A., C. Sun, B. Liu, et X. Wang (2021). Transfer learning and gru-crf augmentation for covid-19 fake news detection. *Computer Science and Information Systems*, 53–53.

Kautz, H. (2022). The Third AI Summer: AAAI Robert S. Engelmore Memorial Lecture. *AI Magazine 43*, 93–104.

Lee, N., B. Z. Li, S. Wang, P. Fung, H. Ma, W. Yih, et M. Khabsa (2021). On unifying misinformation detection. *CoRR abs/2104.05243*.

Lu, Y. et C. Li (2020). GCAN: graph-aware co-attention networks for explainable fake news detection on social media. *CoRR abs/2004.11648*.

Lukasik, M., T. Cohn, et K. Bontcheva (2015). Estimating collective judgement of rumours in social media. *CoRR abs/1506.00468*.

Nguyen, D. Q., T. Vu, A. Rahimi, M. H. Dao, L. T. Nguyen, et L. Doan (2020). WNUT-2020 task 2: Identification of informative COVID-19 english tweets. *CoRR abs/2010.08232*.

Pande, M., A. Budhraja, P. Nema, P. Kumar, et M. M. Khapra (2021). The heads hypothesis: A unifying statistical approach towards understanding multi-headed attention in bert.

Pelrine, K., J. Danovitch, et R. Rabbany (2021). The surprising performance of simple baselines for misinformation detection. *CoRR abs/2104.06952*.

Rogers, A., O. Kovaleva, et A. Rumshisky (2020). A primer in bertology: What we know about how BERT works. *CoRR abs/2002.12327*.

Shu, K., D. Mahudeswaran, S. Wang, D. Lee, et H. Liu (2018). Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *CoRR abs/1809.01286*.

Tay, Y., D. Bahri, D. Metzler, D. Juan, Z. Zhao, et C. Zheng (2020). Synthesizer: Rethinking self-attention in transformer models. *CoRR abs/2005.00743*.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, et I. Polosukhin (2017). Attention is all you need. *CoRR abs/1706.03762*.

Vig, J. (2019). A multiscale visualization of attention in the transformer model. *CoRR abs/1906.05714*.

Vo, N. et K. Lee (2021). Hierarchical multi-head attentive network for evidence-aware fake news detection. *CoRR abs/2102.02680*.

Wang, Y., F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, et J. Gao (2018). Eann: Event adversarial neural networks for multi-modal fake news detection. pp. 849?857.

Zhou, X. et R. Zafarani (2018). Fake news: A survey of research, detection methods, and opportunities. *CoRR abs/1812.00315*.

Summary

Natural Language Processing mainly relies on large language models requiring very large datasets and behaves like a “black box”. These models are the baselines for many classification tasks, such as disinformation detection. Recently, hybrid approaches between deep learning and symbolic AI try to outperform attention-based models by introducing symbolic reasoning in the decision process, making it more understandable to the users. In this paper, we introduce CATS, an explainable attention mechanism based on the semantic analysis of documents. This approach outperforms equivalent fully-neuronal models, reduces annotated data needs and allows to understand how the decision process is made.