

EXTRACTION D'INFORMATIONS

Projet de détection de fake news :

Test de méthodes de preprocessing, de méthodes de vectorisation et d'algorithmes de classification



https://fr.freepik.com/search?format=search&last_filter=query&last_value=fake+nws&query=fake+nws

Réalisé par BESSAC Lucile, SAINTÉ LUCE Constance et SCHLOSSER Guilhem

Année 2023-2024

Sommaire

I. Introduction	3
II. Définitions	3
A. Cadre théorique	3
B. Problématique	3
III. État de l'art	4
IV. Données / Corpus	5
V. Méthodes et expérimentations	7
A. Preprocessing du corpus	7
1. Retrait des Stopwords et de la ponctuation	7
2. Lemmatization	7
B. Méthodes de vectorisation	7
1. CountVectorizer	7
2. TfidfVectorizer	7
3. HashingVectorizer	7
C. Algorithmes de classification	8
1. LinearSVC	8
2. DecisionTreeClassifier	8
3. MultinomialNB	8
4. RandomForestClassifier	8
VI. Evaluation et résultats	9
A. Tableau comparatif des 44 matrices de confusion	9
B. Graphe de comparaison des mesures de précision	10
C. Combinaison présentant les meilleurs résultats	11
D. Combinaison présentant les moins bons résultats	11
E. Un résultat surprenant	12
VII. Perspectives d'amélioration	12
VIII. Difficultés rencontrées pendant le projet	13
IX. Conclusion	13

I. Introduction

Dans le cadre du cours d'Extraction d'Informations du M1 PluriTAL 2023/2024, nous nous sommes penchés sur la tâche de classification automatique de Fake News.

Si l'expression « fake news » naît et se popularise lors de la campagne de Trump 2016 et lors du Brexit, elle est à mettre en parallèle avec l'essor des réseaux sociaux qui jouent un rôle non négligeable dans l'émergence et la diffusion de « fake news ». En effet, les réseaux sociaux ont engendré un changement de comportement chez les utilisateurs dans leur rapport à l'information, dès lors l'internaute devient à la fois récepteur et émetteur. Quant à la signification exacte du terme « fake news » : il est parfois utilisé de manière interchangeable avec des notions telles que rumeurs, théories du complot, propagande et désinformation.

C'est pourquoi nous tenterons de définir dans le cadre de notre projet ce terme qui est traduit en français par infox. Puis, en nous basant sur l'état actuel de la recherche nous nous efforcerons de construire et d'évaluer un outil de détection de « fake news » pour le français le plus précis possible, en testant différentes méthodes de vectorisation et différents algorithmes de classification parmi lesquels : CountVectorizer, TfidfVectorizer, HashingVectorizer et LinearSVC, RandomForestClassifier, MultinomialNB et DecisionTree.

II. Définitions

A. Cadre théorique

Qu'est-ce qu'une « fake news » ? La rumeur peut être vraie ou fausse, cependant une infox par nature est fausse. De plus, si la source d'une rumeur reste inconnue elle est identifiable pour la « fake news ». Alors que les théories du complot sont globales et hégémoniques, les infox sont circonstanciées et événementielles. La propagande est caractérisée par un schéma top-down via les médias traditionnels, tandis que les fausses informations se transmettent davantage dans l'horizontalité.

En bref, les caractéristiques d'une fake news impliquent **la diffusion délibérée, notamment sur l'espace numérique, d'énoncés intentionnellement faux et mensongers par des acteurs sociaux identifiés ou identifiables, qui en assument la responsabilité.**

B. Problématique

Comment utiliser les outils de traitement automatique des langues pour détecter les fausses informations ? Nous aborderons cette question en élaborant un programme de classification d'articles de presse en se basant sur l'analyse textuelle de leur contenu. Nous tenterons de déterminer la meilleure combinaison d'outils pour la classification de fake news dans un corpus d'articles en français..

III. État de l'art

Notre problématique initiale différait de notre problématique actuelle et c'est pourquoi nous avons tout d'abord effectué des recherches afin de mettre en place notre propre programme, qui permettrait de détecter les fake news, plus précisément qui serait en capacité de nous dire à quel point un article est véridique ou faux.

Nous avons étudié les trois articles suivants qui abordent la problématique de la détection des infos et soulignent l'importance croissante de ce domaine de recherche :

Article 1 : [A Survey on Natural Language Processing for Fake News Detection](#)

Article 2 : [Fundamental Theories, Detection Methods, and Opportunities](#)

Article 3 : [Sentiment Analysis for Fake News Detection](#)

Le premier article, "*A Survey on Natural Language Processing for Fake News Detection*", met en avant les avancées et les méthodologies utilisées pour détecter automatiquement les fausses nouvelles. Il insiste sur l'importance de comprendre les facteurs qui influencent l'efficacité des systèmes de détection et propose des recommandations pour les recherches futures.

Le deuxième article, "*Fundamental Theories, Detection Methods, and Opportunities*", se concentre sur une approche multidimensionnelle de la détection des fausses nouvelles, en évaluant les connaissances, le style, la diffusion et la source de l'information.

Enfin, le troisième article, "*Sentiment Analysis for Fake News Detection*" publié dans le journal "Electronics", montre l'utilisation de l'analyse des sentiments pour la détection des fake news, en mettant en évidence son importance dans un contexte de d'augmentation des partages des fausses nouvelles sur les médias sociaux.

Ces trois articles donnent un aperçu des défis et des opportunités dans le domaine qui contribuent à faire diminuer la désinformation et à apporter un environnement d'information fiable et transparent.

Suite à l'étude de ces trois articles, nous avons en tête de mettre au point et comparer 3 méthodes :

1. Méthode basée sur les connaissances
2. Méthode basée sur le style
3. Méthode basée sur les sources

Nous nous sommes finalement aperçu que la tâche était vaste et avons préféré nous concentrer sur l'évaluation de méthodes de vectorisation et de classification préexistantes. Nous avons donc choisi de tester en priorité les méthodes citées dans nos articles comme la vectorisation TF-IDF et la classification avec un SVM et nous avons comparé ces méthodes à d'autres méthodes.

Finalement, nos recherches sur l'état de l'art nous ont surtout permis d'envisager des perspectives d'amélioration pour de futurs travaux.

IV. Données / Corpus

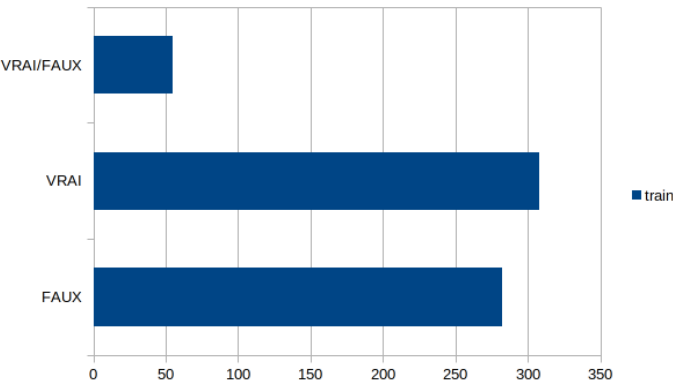
Nos recherches de corpus annoté pour la classification de fake-news en français sur les différentes plateformes telles que The Huggingface ou Kaggle se sont révélées infructueuses, c'est pourquoi nous avons décidé de constituer nous même notre corpus d'article de presse labellisé.

À l'aide des bibliothèques request et BeautifulSoup nous avons appliqué les méthodes du scrapping (voir : `scrapping.py` du dossier Script) sur le site : <https://www.hoaxbuster.com/>. Suite à cette première étape, nous avons pu récupérer quelques 441 articles avec leur métadonnées tel que le titre, le sous titre, l'url, l'auteur, la date, le label et la catégorie de l'article. Cependant, 324 articles sur 441 sont étiquetés comme 'Faux' tandis que seulement 47 sont labellisés 'Vrai', 61 sont marqués comme 'Du vrai / du faux' et seulement 9 pour 'Analyse en cours' (voir : tableau 1). Ce déséquilibre constitue un obstacle pour l'entraînement de l'algorithme de classification. Cette situation risque en effet de provoquer un surapprentissage sur les articles marqués comme faux, entraînant ainsi une tendance de l'algorithme à classer incorrectement certains articles comme étant faux alors qu'ils sont vrais.

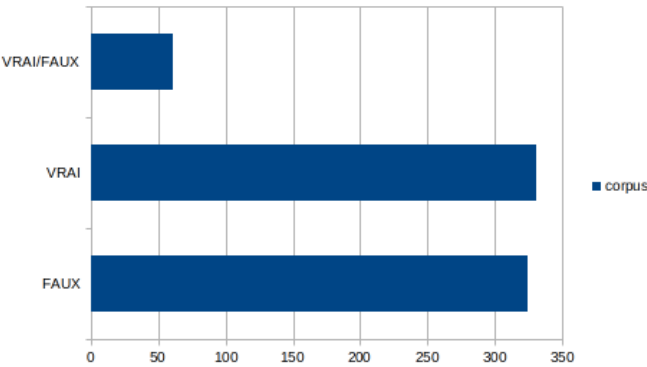
C'est pourquoi dans un second temps nous avons implémenté un autre script python (voir fichier : `enrichissement_corpus.py`), pour enrichir notre corpus annoté d'articles étiquetés comme vrais et supprimer les articles étiqueté 'Analyse en cours' car non pertinent pour notre analyse. Pour cela nous avons récupéré les fichiers textes des articles de presse présents sur le drive du cours. Ainsi notre corpus enrichie compte 331 articles 'Vrai', 324 'Faux', 61 'Du vrai / du faux' et 0 'Analyse en cours' (voir : tableau 1 et graph 1)

	Vrai	Faux	Du vrai / du faux	Analyse en cours	total
data.json	47	324	61	9	441
data_enrichi.json	331	324	61	0	716
train.json	308	282	55	0	645
test.json	23	42	6	0	71

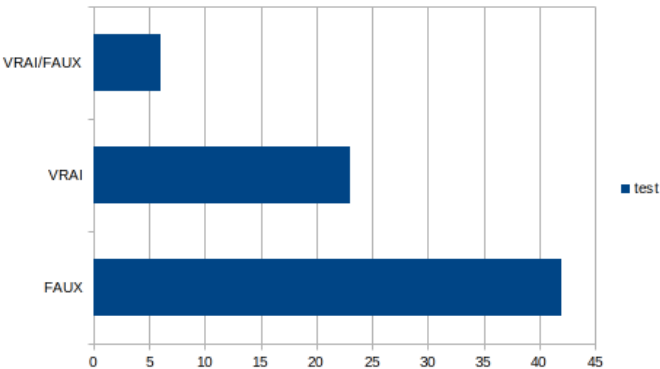
Tableau 1 : Répartition des articles par labels en fonction des fichiers json



Graphe 1 : Répartition des articles par labels sur data_enrichi.json



Graphe 2 : Répartition des articles par label sur train.json



Graphe 3 : Répartition des articles par labels sur test.json

V. Méthodes et expérimentations

Afin d'élaborer le meilleur outil de classification de fake news pour le français, nous avons divisé le processus en 3 étapes : Preprocessing du corpus, Vectorisation, Classification. Nous avons ensuite comparé différentes méthodes pour chaque étape, dans le but de définir la meilleure combinaison.

A. Preprocessing du corpus

1. Retrait des Stopwords et de la ponctuation

Les stopwords sont des mots couramment utilisés dans une langue mais ne portant généralement que peu de sens pour une analyse linguistique. Nous avons fait l'hypothèse que supprimer ces stopwords améliorerait les résultats en augmentant le poids des autres mots.

2. Lemmatization

Le lemme d'un mot est sa forme non fléchi (ni en genre ni en nombre pour le français), autrement dit la forme sous laquelle le terme apparaît dans le dictionnaire. Nous avons supposé que lemmatiser notre texte améliorerait les résultats en donnant plus de poids à un même mot.

B. Méthodes de vectorisation

1. CountVectorizer

Nous avons réalisé différentes combinaisons avec "**CountVectorizer**", qui vectorise de manière à compter le nombre d'occurrence de chaque mot dans un texte. Cependant, il peut accorder trop d'importance aux termes qui apparaissent fréquemment, sans se préoccuper de la pertinence des mots dans le contexte du corpus.

2. TfidfVectorizer

Le vectorizer, "**TfidfVectorizer**", combine la fréquence d'un mot dans un document avec la fréquence inverse de ce même mot dans le corpus. Le résultat est représenté sous forme de matrice. Ce qui est intéressant avec ce vectorizer, comparé à son homologue, est qu'il donne moins d'importance aux mots qui se répètent fréquemment, comme les stopwords par exemple.

3. HashingVectorizer

HashingVectorizer, est un énième vectorizer qui permet de faire des matrices de confusion. Comme son nom l'indique, il réalise un hachage des données, grâce à une fonction (MurmurHash3) qui convertit le corpus utilisé afin d'en faire une matrice de token. Ce vectorizer est efficace en ce qui concerne la mémoire et le traitement de grand corpus. L'inconvénient cependant, est que l'on peut perdre des données, si 2 mots ont la même taille de dimension dans le vecteur, le vectorizer ne fera pas de distinction entre les 2.

C. Algorithmes de classification

1. LinearSVC

Le classificateur **LinearSVC** cherche les hyperplans afin de les séparer en sous-ensembles linéairement. Il utilise les machines à vecteur de support, aussi communément appelé "SVM", des algorithmes qui réalisent des tâches de classification et permet d'éviter le surapprentissage. Ce classificateur est efficace pour s'entraîner sur des données de grande dimension, faire de la prédiction. En revanche, il peut être difficile à interpréter car l'hyperplan peut rendre cette tâche difficile. Il n'est pas conseillé de l'utiliser lorsque les jeux de données sont trop grands et si les données ne sont pas linéairement séparables.

2. DecisionTreeClassifier

DecisionTreeClassifier, permet comme indiquer dans son nom d'utiliser des arbres de décision, ils permettent de partitionner des données en sous-ensembles. Pour utiliser ce classificateur, il n'est pas nécessaire de faire une normalisation des données.

3. MultinomialNB

Un **classificateur bayésien naïf**, qui traite des données multinomiales. Il peut être utilisé pour la classification de texte. Une donnée multinomiale est une généralisation de la distribution binomiale pour les variables qui détiennent plus de deux modes possibles.

4. RandomForestClassifier

Tout comme le classificateur précédent, **RandomForestClassifier** utilise des arbres de décision (une forêt). Cette fois, chaque arbre de décision traite une sous-ensemble aléatoire de données et des caractéristiques. Cette classification permet de réduire le risque de surapprentissage, comparé à un seul arbre. Il est par conséquent plus fastidieux à entraîner et à comprendre.

VI. Evaluation et résultats

Comme nous l'avons précédemment exposé, l'objectif de ce rapport est de déterminer quelle est la meilleure combinaison de méthode de preprocessing, méthode de vectorisation et méthode de classification. Ainsi, nous avons pu générer 44 combinaisons et avons choisi de présenter celles qui nous semblaient particulièrement intéressantes.

A. Tableau comparatif des 44 matrices de confusion



Figure 1: Tableau comparatif des 44 combinaisons testées.

Bien qu'il ne soit pas des plus précis, ce tableau nous permet en un coup d'œil d'avoir une idée de quelles combinaisons ont de bons résultats ou pas.

Il nous permet aussi de nous rendre compte que dans la majorité des cas, le retrait ou non des stopwords semble ne pas avoir de grande incidence sur les résultats.

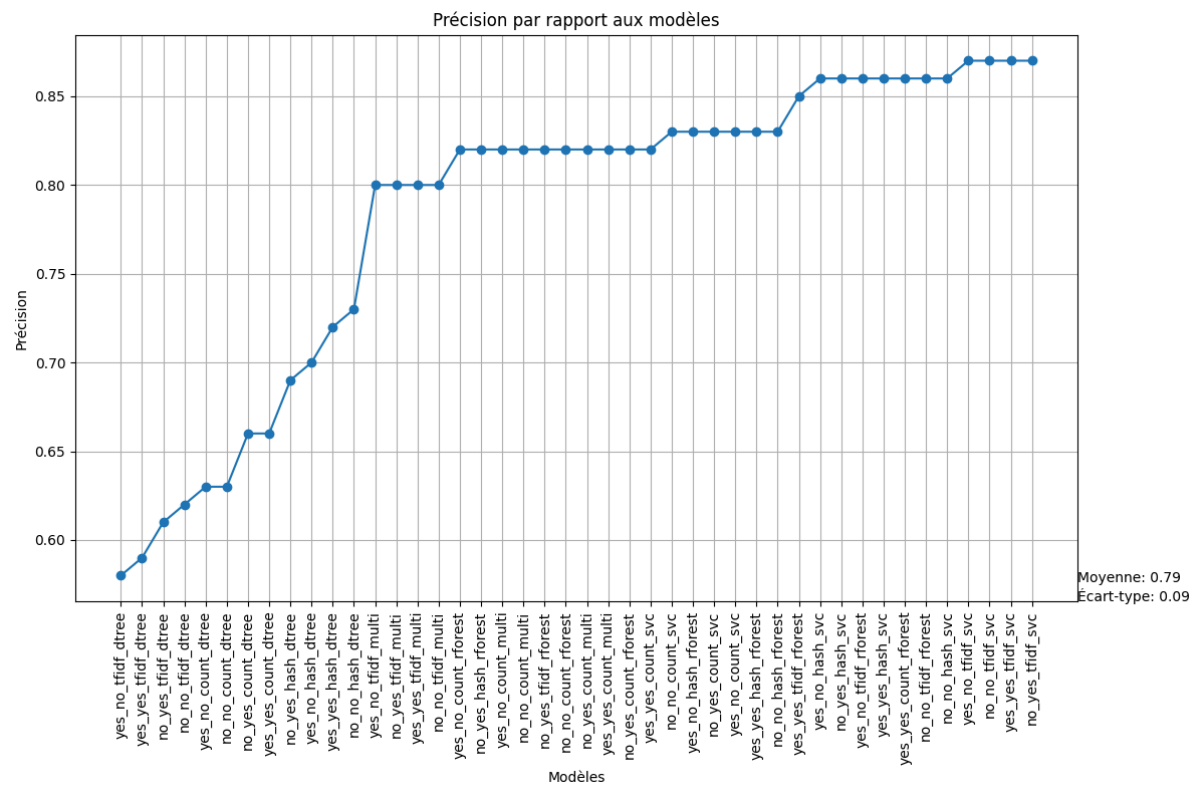
Chose étonnante, il semblerait que la lemmatisation empire les résultats de l'algorithme de classification DecisionTree. Cela doit avoir une explication mathématique et doit être lié à la manière dont le vectoriseur et l'algorithme de classification fonctionnent et il serait intéressant de l'étudier dans un futur travail.

En comparant les matrices, il apparaît clairement qu'aucune combinaison ne permet de classer correctement les textes étiquetés "du vrai / du faux". Cela tient sûrement à plusieurs choses dont : Le fait que le corpus de train n'est pas équilibré puisqu'il ne contient

que 61 articles de ce type contre 331 et 324 pour les autres catégories. Ainsi, le modèle ne peut pas apprendre correctement sur ce type d'articles. De plus, un article étiqueté "du vrai / du faux" contient par essence un mélange des caractéristiques des articles contenant des infos et de ceux contenant des infox. Il semble donc normal que les erreurs de classifications soient plus fréquentes sur cette catégorie.

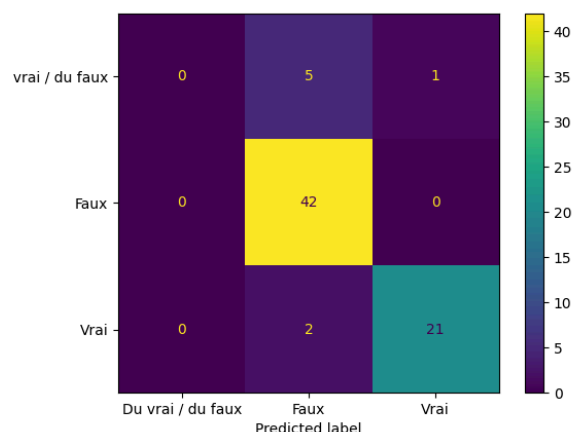
B. Graphe de comparaison des mesures de précision

Dans un souci de clarté nous avons établi un graphique (voir Graphique 1) qui répertorie tous les scores de précision rangés par ordre croissant. Ainsi les modèles les plus à gauche du graphique sont ceux qui performant le moins bien tandis que les modèles les plus à droite sont les plus performant pour la détection de fake news. Le nom des modèles sur l'axe des x du graphique 1 sont défini selon le pattern suivant : preprocess_lemmatizer_vectorizer_model, par exemple pour le premier modèles le plus à gauche : yes_no_tfidf_dtrees. Cela signifie que le prétraitement (traitement des stopwords) a été appliqué (yes), la lemmatisation non (no), le choix du vectorizer est TfidfVectorizer, et le modèle choisi est DecisionTree. Bien que les résultats varient légèrement à chaque exécution du code, ils restent généralement cohérents dans leur tendance globale. On observe que l'application des options preprocess et lemmatizer a peu d'impact sur la précision. En revanche, le paramètre ayant la plus grande influence sur la précision est le type de modèle utilisé, en effet on constate que l'intégralité des modèles DecisionTree sont à gauche du graphe tandis que la majorité des modèles LinearSVC sont à droite du graphe.



Graphique 1 : Précision en fonction des modèles

C. Combinaison présentant les meilleurs résultats



La combinaison ayant obtenu les meilleurs résultats est la suivante : retrait des stopwords, lemmatisation, vectorisation avec TF-IDF Vectorizer et Classification avec LinearSVC.

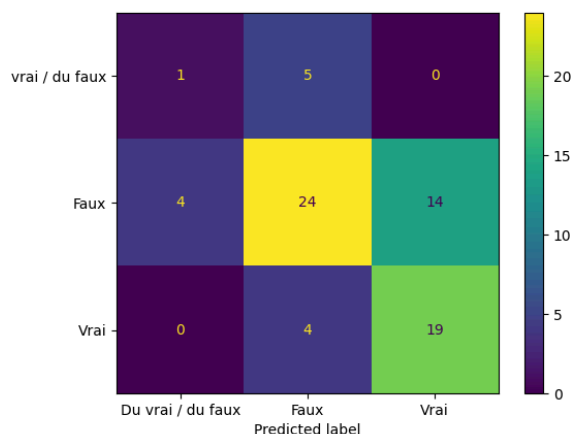
C'est donc la combinaison que nous conseillons d'utiliser pour la détection de fake news en français.

Figure 2 : Matrice de confusion de la combinaison *tfidf-svc-preprocessed-lem*

	Précision	Rappel	F1-Score	Support
Du vrai / du faux	0.00	0.00	0.00	6
Faux	0.86	1.00	0.92	42
Vrai	0.95	0.91	0.93	23
accuracy			0.89	71
macro avg	0.60	0.64	0.62	71
weighted avg	0.82	0.89	0.85	71

Table 1: Rapport de classification de la combinaison *tfidf-svc-preprocessed-lem*

D. Combinaison présentant les moins bons résultats



La combinaison ayant obtenu les moins bons résultats est la suivante : pas de retrait des stopwords, pas de lemmatisation, vectorisation avec HashVectorizer et Classification avec DecisionTree.

C'est donc la combinaison que nous ne conseillons pas d'utiliser pour la détection de fake news en français.

Figure 3 : Matrice de confusion de la combinaison *hash-dtree*

	Précision	Rappel	F1-Score	Support
Du vrai / du faux	0.20	0.17	0.18	6
Faux	0.73	0.57	0.64	42
Vrai	0.58	0.83	0.68	23
accuracy			0.62	71
macro avg	0.50	0.52	0.50	71
weighted avg	0.63	0.62	0.61	71

Table 2: Rapport de classification de la combinaison *hash-dtree*

E. Un résultat surprenant

À plusieurs reprises, il semblerait que la lemmatisation empire les résultats de l’algorithme de classification DecisionTree. Cela doit avoir une explication mathématique et doit être lié à la manière dont le vectoriseur et l’algorithme de classification fonctionnent et il serait intéressant de l’étudier dans un futur travail.

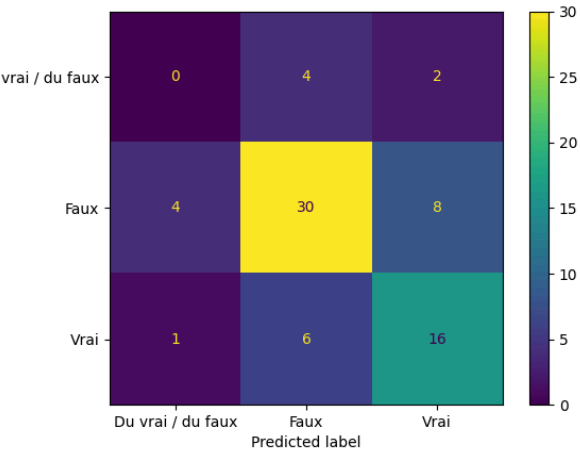


Figure 4 : Matrice de confusion de la combinaison

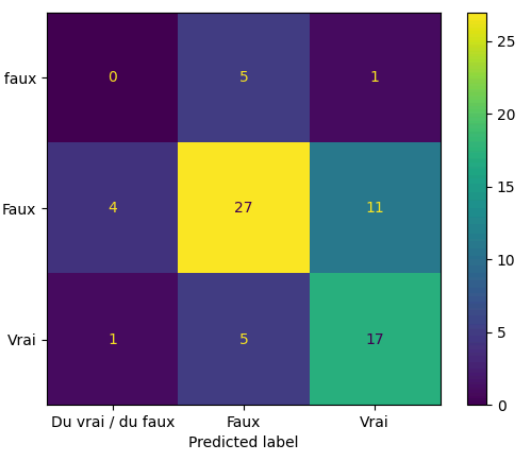


Figure 5 : Matrice de confusion de la

	Précision	Rappel	F1-Score	Support
Du vrai / du faux	0.00	0.00	0.00	6
Faux	0.75	0.71	0.73	42
Vrai	0.62	0.70	0.65	23
accuracy			0.65	71
macro avg	0.46	0.47	0.46	71
weighted avg	0.64	0.65	0.64	71

Table 4: Rapport de classification de la combinaison count-dtree-preprocessed

	Précision	Rappel	F1-Score	Support
Du vrai / du faux	0.00	0.00	0.00	6
Faux	0.73	0.64	0.68	42
Vrai	0.59	0.74	0.65	23
accuracy			0.62	71
macro avg	0.44	0.46	0.45	71
weighted avg	0.62	0.62	0.62	71

Table 5: Rapport de classification de la combinaison count-dtree-preprocessed-lem

count-dtree-preprocessed

combinaison count-dtree-preprocessed-lem

VII. Perspectives d’amélioration

Dans un premier temps, pour améliorer nos résultats il faudrait écarter les articles dont le contenu est vide. En effet, nous avons remarqué qu’une cinquantaine d’articles issue du scrapping ont un contenu dont le nombre de token ne dépasse pas la longueur du titre et du sous-titre cumulé. Généralement, ce sont des articles réalisés en collaboration avec d’autres médias, dont le contenu se résume souvent à l’adresse URL menant vers l’article en question. De plus, il serait envisageable de faire de la data augmentation pour le label : “Du vrai / du faux” qui concerne les articles contenant des éléments factuels mais présenté de façon trompeuse.

Dans un second temps, nous pourrions envisager un projet plus ambitieux. À partir d’une adresse URL, l’algorithme récupérerait le contenu de l’article ainsi que ses métadonnées. En se basant sur le contenu de l’article, un premier indicateur de fiabilité serait généré à l’aide de notre meilleur modèle et d’un corpus plus riche et homogène. De plus, une analyse de sentiment pourrait être réalisée afin de déterminer si le contenu suscite

un fort engagement émotionnel, ce qui pourrait signaler une possible infox ou un contenu généré pour attirer des clics. Enfin dernier paramètre, en examinant les métadonnées, nous pourrions identifier l'auteur et le site hébergeant l'article pour évaluer la crédibilité des sources. En faisant la moyenne de nos trois paramètres, on obtiendrait ainsi une prédiction plus aboutie.

Enfin, il serait intéressant de mettre en place une technique fiable pour comparer précisément les performances de chacune des 44 combinaisons.

VIII. Difficultés rencontrées pendant le projet

Notre première grande difficulté a résidé dans le fait que nous étions d'abord partis sur la création de A à Z de méthodes de classification et avons en tête de mettre au point et comparer 3 méthodes :

4. Méthode basée sur les connaissances
5. Méthode basée sur le style
6. Méthode basée sur les sources

Nous nous sommes finalement aperçu que la tâche était vaste et avons préféré nous concentrer sur l'évaluation de méthodes de vectorisation et de classification préexistantes.

La constitution d'un corpus de fake news s'est révélée problématique. Bien qu'un certain nombre de ressources soit disponible en ligne pour la langue anglaise nous n'avons pas trouvé de corpus pour le français. C'est pourquoi nous avons constitué un corpus annoté en scrappant le site web : <https://www.hoaxbuster.com/> (voir : répertoire Script fichier scrapping.py).

Côté code, bien que nous ayons initialisé une graine pour la reproductibilité des résultats aléatoires, nous avons été confrontés à des résultats du même ordre mais légèrement différents à chaque fois que l'on lance le programme. Ce comportement est peut-être dû à au calcul effectué par les différents vectoriseur et modèle, cependant à ce stade nous ne sommes pas en mesure de déterminer l'origine de ce comportement.

IX. Conclusion

Bien que la production et la diffusion d'informations mensongères fassent partie intégrante de l'histoire de l'humanité, le concept de fake news est étroitement lié au développement de l'espace numérique et notamment des réseaux sociaux. Car on sait que la vitesse de propagation des infox est supérieure aux autres informations sur ces mêmes plateformes. En effet, les réseaux sociaux ont une propension à mettre en avant les contenus viraux dans le but de générer du trafic et dégager des revenus. Par conséquent, elles se retrouvent prises au piège entre deux impératifs, l'un économique et l'autre éthique. Face au flux d'informations le nlp peut jouer un rôle de premier plan dans la détection de fake news, c'est pourquoi, nous nous sommes lancés dans ce projet. Lors de nos recherches nous avons constaté le peu de ressource disponible pour la détection de

fake-news en langue française. Néanmoins en constituant notre propre corpus annoté nous avons pu évaluer différents modèles de classification. D'après nos résultats, l'utilisation du `tfidfVectorizer` combiné au classificateur `LinearSVC` est la plus adéquate pour cette tâche. Pour améliorer la précision de notre modèle, il faudrait retirer les articles vides du corpus et l'enrichir davantage en équilibrant les différents labels et catégories.

Références

Articles de l'état de l'art

- Ray Oshikawa, Jing Qian, William Yang Wang, "A Survey on Natural Language Processing for Fake News Detection", Cornell University, 2018, <https://arxiv.org/abs/1811.00770>
- Reza Zafarani, Xinyi Zhou, "A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities", ACM Digital Library, 2020, <https://dl.acm.org/doi/10.1145/3395046>
- Miguel A. Alonso, David VILARES, Carlos Gómez-Rodríguez, et Jesús Vilares, "Sentiment Analysis for Fake News Detection", MDPI, electronics, 2021, <https://www.mdpi.com/2079-9292/10/11/1348>

Articles Cairn pour la définition de fake news

- Julien Giry, "Les *fake news* comme concept de sciences sociales", Cairn, 2020, <https://www.cairn.info/revue-questions-de-communication-2020-2-page-371.htm>
- Florian Dauphin, "Les *Fake News* au prisme des théories sur les rumeurs et la propagande", 2019, <https://journals.openedition.org/edc/9132>
- Stéphane Dangel, "Rosa Cetro et Lorella Sini (dirs), *Fake news rumeurs, intox... Stratégies et visées discursives de la désinformation*", coll. Humanités numériques, 2021, <https://www.cairn.info/revue-questions-de-communication-2021-2-page-553.htm>
- Marc Riglet, Une histoire de *fake news*, Cairn, 2019, <https://www.cairn.info/revue-humanisme-2019-2-page-1.htm>
- Lara Quijano-Sanchez, Federico Liberatore, José Camacho-Collados et Miguel Camacho-Collados, "Applying automatic text-based detection of deceptive language to police reports: extracting behavioral patterns from a multi-step classification model to understand how we lie to the Police", Cardiff University, 2018, https://orca.cardiff.ac.uk/id/eprint/113133/1/VeriPol_KnowledgeBased.pdf
- Pierre Vandeginste, "Repérer les circuits de la désinformation...", data analytics post 2021, <https://dataanalyticspost.com/reperer-circuits-desinformation/>

Articles scientifiques

- Ray Oshikawa, Jing Qian, William Yang Wang, "A Survey on Natural Language Processing for Fake News Detection", Research Gate, 2018, https://www.researchgate.net/publication/328736618_A_Survey_on_Natural_Language_Processing_for_Fake_News_Detection

- Maximilian Nickel, Kevin Murphy, Volker Tresp, Evgeniy Gabrilovich, "A Review of Relational Machine Learning for Knowledge Graphs" , Cornell University, 2015, <https://arxiv.org/abs/1503.00759> (graphe de connaissance)
- Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Gerhard Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia", Science Direct, 2013, <https://www.sciencedirect.com/science/article/pii/S0004370212000719> (graphe de connaissance YAGO2)
- Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, Wei Zhang, "Knowledge vault: a web-scale approach to probabilistic knowledge fusion", ACM Digital Library, 2014, <https://dl.acm.org/doi/10.1145/2623330.2623623> (graphe de connaissance LinkNBed)
- Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, "Learning with Local and Global Consistency", Research Gate, 2004, https://www.researchgate.net/publication/2869065_Learning_with_Local_and_Global_Consistency (analyse du style)
- Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea, "Automatic Detection of Fake News", ACL Anthology, 2018, <https://aclanthology.org/C18-1287/> (analyse du style)
- Miguel A. Alonso, David VILARES, Carlos Gómez-Rodríguez, et Jesús Vilares, "Sentiment Analysis for Fake News Detection", MDPI, electronics, 2021, <https://www.mdpi.com/2079-9292/10/11/1348> (analyse de sentiment)
- Niraj Sitaula, Chilukuri K. Mohan, Jennifer Grygiel, Xinyi Zhou, Reza Zafarani, "Credibility-based Fake News Detection", Cornell University, 2019, <https://arxiv.org/abs/1911.00643> (analyse source auteur)
- Varol, O., Ferrara, E., Davis, C., Menczer, F., & Flammini, A. (2017). Online Human-Bot Interactions: Detection, Estimation, and Characterization. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1), 280-289. <https://ojs.aaai.org/index.php/ICWSM/article/view/14871> (analyse source utilisateur)