

UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des sciences

Département d'informatique



MÉMOIRE DE MASTER

En Informatique

Option : Ingénierie Des Logiciels

THÈME :

Profilage des Influenceurs de Cryptomonnaies sur
Twitter

Réalisé par :

ALLAG Maroua

BOULHOUACHE Roumaissa

Encadré par :

Mme. AMMAR-AOUCHICHE Imene

Mr. REMMIDE Mohamed Abdelkarim

juillet 2024

Remerciements

Avant toute chose, nous souhaitons exprimer notre reconnaissance envers Allah, le Tout-Puissant, qui nous a accordé la force et la patience nécessaires pour mener à bien ce travail modeste.

Nous adressons nos sincères remerciements à notre promotrice, **Mme. AMMAR-AOUCHICHE Imene**, pour avoir guidé nos efforts et nous avoir éclairés tout au long de ce chemin. Ses conseils précieux ont joué un rôle crucial dans l'amélioration de ce mémoire, et nous saluons sa dévotion et sa disponibilité pendant toute la durée de notre projet, ainsi que le temps précieux qu'il nous a consacré.

Nous exprimons nos sincères remerciements à **Mme. BOUMAHDJ** pour son accueil chaleureux, la proposition de ce sujet, l'opportunité qu'elle nous a offerte pour l'approfondir, et la confiance qu'elle nous a accordée.

Nous exprimons également notre gratitude envers les membres du jury, dont la présence nous a honorés et qui ont accepté de juger notre travail.

Nous tenons à remercier chaleureusement notre encadreur, **Mr. REMMIDE Mohamed Abdelkarim**, pour son accompagnement et ses encouragements tout au long de ce parcours.

Enfin, nos remerciements les plus sincères vont à nos familles, véritables piliers de cette réussite, pour leur soutien indéfectible et leurs encouragements constants qui nous ont permis de mener ce travail à terme avec succès.

Dédicaces

Je tiens tout d'abord à exprimer ma profonde gratitude envers **ma mère**, qui a assumé avec courage et dévouement les rôles essentiels de mère et de père dans ma vie. Ton soutien inconditionnel et ta force m'ont toujours guidée et inspirée, et je te suis infiniment reconnaissante pour tout ce que tu as sacrifié pour mon éducation et mon bien-être.

Un merci particulier à mes deux frères, **Mohamed Amine** et **Ali**, qui ont été à mes côtés à chaque étape de ce parcours. Votre soutien et vos conseils avisés ont été précieux et m'ont permis d'avancer avec confiance.

À ma belle-sœur **Wissam**, qui m'a toujours soutenue et encouragée sans relâche, je t'adresse toute ma reconnaissance. Ta présence et ton soutien ont été une source constante de réconfort et de motivation.

Je souhaite également exprimer ma gratitude à mes meilleures amies **Asma, Romaisa, Louiza, Maissa** et à tous ceux et celles qui m'ont accompagnée. Votre amitié sincère et votre soutien indéfectible ont illuminé mon parcours et m'ont donné la force nécessaire pour avancer.

À l'âme de ma grand-mère, qui a toujours été comme une douce mère. J'aurais tellement aimé que tu sois là pour partager ce moment important de ma vie, mais Allah en a décidé autrement. Que ton âme repose en paix.

à mon neveu **Amjed Adem**, dont le sourire a su illuminer même mes pires jours : sache que ta tante t'aime infiniment. Que Dieu te protège et te guide toujours.

Merci aussi à mon ami **Ilyes**.

Merci à chacun d'entre vous pour votre amour, votre soutien et votre présence inestimable tout au long de ce chemin vers la réussite. Vous êtes mes piliers et ma force.

Maroua

Dédicaces

Je tiens tout particulièrement à exprimer ma profonde gratitude envers **ma mère** et **mon père** pour leur sagesse inébranlable, leur infinie patience et leur amour indéfectible qui ont toujours été une source constante de motivation et de réconfort pour moi. Leur confiance absolue en mes capacités m'a toujours inspiré et m'a donné la force nécessaire pour surmonter les défis et persévérer dans mes études. L'exemple de leur éthique de travail exemplaire a été une source d'inspiration et un modèle pour moi. Leur accompagnement continu et leurs conseils avisés ont joué un rôle crucial dans mon parcours et ont grandement contribué à ma réussite.

À mes chères belles-sœurs **Imene** et **Amira**, votre soutien constant et votre affection ont toujours été précieux pour moi. Vos encouragements réguliers et votre présence bienveillante ont joué un rôle essentiel dans l'équilibre sain entre ma vie professionnelle et personnelle.

À mon cher frère **Abdelillah**, ta vivacité d'esprit et ton enthousiasme ont été une source d'inspiration constante pour moi. Ta présence et ton soutien ont été d'une importance inestimable tout au long de cette étude.

Je souhaite également exprimer ma reconnaissance envers mes meilleures amies **Zineb** et **Kholoud, Marouâ**, ainsi qu'à tous ceux et celles qui m'ont soutenue. Leur amitié sincère et leur soutien constant ont été des éléments fondamentaux qui ont illuminé ma route et m'ont motivée à progresser.

À ma précieuse nièce **Maram**, qui occupe une place spéciale dans mon cœur, je prie sincèrement pour que Dieu veille sur toi, te protège et t'accorde une longue vie. Ton sourire a toujours été une source continue d'inspiration pour moi, m'apportant la patience indispensable pour faire face à chaque jour.

Je souhaite exprimer ma profonde gratitude à toute ma famille pour votre soutien précieux, votre amour inébranlable et votre présence constante dans ma vie. Chacun de vous a joué un rôle essentiel en apportant joie, réconfort et encouragement à chaque étape de mon parcours. Votre soutien a été crucial pour mes succès et m'a donné la résilience nécessaire pour faire face aux défis. Je vous suis profondément reconnaissante et je chéris chaque moment passé ensemble.

Je vous suis reconnaissant pour votre amour constant, votre soutien inestimable et votre présence indispensable tout au long de mon parcours vers la réussite. Vous êtes ma source de stabilité et de motivation.

Roumaissa

Résumé

Au cours des dernières années, les cryptomonnaies ont profondément transformé le secteur financier grâce à la technologie de la blockchain, permettant des transactions sécurisées et décentralisées. Cette évolution a engendré une nouvelle génération d'influenceurs spécialisés, jouant un rôle crucial dans l'éducation du public, l'analyse des tendances du marché, la promotion de projets innovants et la gestion de communautés en ligne. Toutefois, cette montée en puissance des influenceurs a aussi posé des défis importants, tels que la propagation de désinformation, les conflits d'intérêts et les fraudes, contribuant ainsi à accroître la volatilité des prix et mettant en évidence la nécessité d'une régulation renforcée pour restaurer la confiance des investisseurs. Les médias sociaux, quant à eux, ont amplifié ces problèmes en facilitant la diffusion de désinformation et la manipulation du marché, soulignant l'importance de mécanismes de vérification et de surveillance rigoureuse.

Dans ce mémoire, nous avons exploré une solution à ces défis en nous concentrant sur le profilage des influenceurs dans le domaine des cryptomonnaies. Notre méthode comprenait trois étapes principales : évaluer leur degré d'influence, comprendre leurs intérêts spécifiques et examiner leurs intentions.

Pour ce faire, nous avons utilisé deux modèles d'apprentissage par transfert, SetFit et DistilBERT, appliqués au jeu de données Twitter PAN2023 avec un nombre limité d'itérations d'apprentissage.

Nous avons mené plusieurs expériences sur ce jeu de données et avons atteint un taux de précision de 80,77% avec le modèle DistilBERT et de 82,69% avec le modèle SetFit pour le profilage des influenceurs de cryptomonnaie.

Mots clés :

SetFit, DistilBERT, Apprentissage par transfert, Profilage des influenceurs de cryptomonnaie.

Abstract

In recent years, cryptocurrencies have profoundly transformed the financial sector through blockchain technology, enabling secure and decentralized transactions. This evolution has led to the rise of a new generation of specialized influencers who play a crucial role in educating the public, analyzing market trends, promoting innovative projects, and managing online communities. However, this growing influence has also posed significant challenges, such as the spread of misinformation, conflicts of interest, and fraud, contributing to increased price volatility and highlighting the need for strengthened regulation to restore investor confidence. Social media, in turn, has amplified these issues by facilitating the spread of misinformation and market manipulation, underscoring the importance of robust verification and monitoring mechanisms.

In this thesis, we explored a solution to these challenges by focusing on profiling influencers in the cryptocurrency domain. Our method consisted of three main steps : evaluating their level of influence, understanding their specific interests, and examining their intentions. To achieve this, we used two transfer learning models, SetFit and DistilBERT, applied to the PAN2023 dataset with a few-shot learning.

Through various experiments on this dataset, we achieved an accuracy rate of 80.77% with the DistilBERT model and 82.69% with the SetFit model for cryptocurrency influencer profiling.

Keywords :

SetFit, DistilBERT, Author profiling, Transfer learning, Cryptocurrency influencers.

الملخص

في السنوات الأخيرة، أحدثت العملات المشفرة تحولاً عميقاً في القطاع المالي بفضل تقنية Blockchain، مما أتاح إجراء معاملات آمنة ولا مركزية. وقد أسفر هذا التطور عن ظهور جيل جديد من المؤثرين المتخصصين الذين يلعبون دوراً حيوياً في تعليم الجمهور، وتحليل اتجاهات السوق، والترويج للمشاريع المبتكرة، وإدارة المجتمعات عبر الإنترنت. ومع ذلك، فقد طرحت هذه القوة المتزايدة للمؤثرين تحديات مهمة، مثل انتشار المعلومات المضللة، وتضارب المصالح، والاحتمالات، مما ساهم في زيادة تقلبات الأسعار وأبرز الحاجة إلى تعزيز التنظيم لاستعادة ثقة المستثمرين. بدورها، فاقتت وسائل التواصل الاجتماعي هذه المشكلات من خلال تسهيل نشر المعلومات المضللة والتلاعب بالسوق، مما يؤكد على أهمية وجود آليات للتحقق والمراقبة الصارمة.

في هذه الرسالة، استكشفنا حلاً لهذه التحديات من خلال التركيز على تصنيف المؤثرين في مجال العملات المشفرة. تضمنت طريقتنا ثلاث خطوات رئيسية : تقييم مدى تأثيرهم، فهم اهتماماتهم المحددة، واستكشاف نواياهم. لتحقيق ذلك، استخدمنا نموذجين للتعليم بالنقل SetFit و DistilBERT مطبقين على مجموعة بيانات تويتر PAN2023 مع عدد محدود من دورات التعلم. لقد أجرينا العديد من التجارب على مجموعة البيانات هذه وحققنا معدل دقة قدره 80.77% باستخدام نموذج DistilBERT و 82.69% باستخدام نموذج SetFit لتصنيف الشخصيات المؤثرة في العملات المشفرة.

الكلمات المفتاحية :

DistilBERT, SetFit, التعلم النقلي، تحديد ملاح المؤثرين، تصنيف المؤثرين في مجال العملات المشفرة.

Table des matières

Table des figures

Liste des tableaux

Introduction Générale	1
Contexte de travail	1
Problématique	1
Objectifs du travail	2
Organisation du mémoire	2
1 Profilage des auteurs	4
1.1 Introduction	4
1.2 Traitement du langage naturel (TLN)	4
1.3 Analyse des auteurs	5
1.4 Les catégories d'analyse des auteurs	5
1.4.1 Attribution des auteurs	5
1.4.2 Vérification des auteurs	6
1.4.3 Profilage des auteurs	7
1.5 Travaux connexes	9
1.5.1 Profilage des auteurs	9
1.5.2 Cryptomonnaies	11
1.6 Discussion	15
1.7 Conclusion	16
2 Solution proposée	17
2.1 Introduction	17
2.2 Architecture globale	17
2.3 L'ensemble de données	20
2.4 Préparation des données	20

2.5	Prétraitement	21
2.6	Ensemble Entraînement/Test	23
2.7	Construction des modèles	24
2.8	Les modèles Transformer	24
2.8.1	SetFit	25
2.8.1.1	Transformateur de phrase (Sentence-transformers)	27
2.8.1.2	Affiner le transformateur de phrases pré-entraîné (Sentence Transformer Fine-Tuning)	27
2.8.1.3	Entraînement de la tête de classification (Classification Head Training)	29
2.8.2	DistilBERT	30
2.8.2.1	Chargement du modèle	32
2.8.2.2	Tokenisation des tweets à l'aide du tokenizer "Distilbert"	32
2.8.2.3	Mise au point du modèle (Fine-tuning of the model)	33
2.8.2.4	Le générateur des représentation latente (The latent representation generator)	33
2.8.2.5	Formation des classificateurs (Training of classifiers)	33
2.9	Conclusion	34
3	Expérimentation et résultats	35
3.1	Introduction	35
3.2	Environnement du logiciel	35
3.2.1	Kaggle	35
3.2.2	Bibliothèques Python	36
3.2.2.1	NumPy	36
3.2.2.2	Pandas	36
3.2.2.3	Scikit-learn	36
3.2.2.4	Transformers	36
3.3	Mesures d'évaluation	37
3.3.1	Accuracy	37
3.3.2	F1-score	38
3.4	Configuration expérimentale	38
3.4.1	Tweets purs vs tweets prétraités	38
3.4.1.1	Utilisation des tweets bruts	38
3.4.1.2	Utilisation des tweets prétraités	43
3.4.2	Analyse comparative des performances des deux modèles	48
3.4.3	Comparaison des résultats avec l'état de l'art	49
3.5	Interface d'application	50
3.6	Conclusion	52

Conclusion et perspectives	53
Bibliographie	55

Table des figures

1.1	Présentation des catégories de l'analyse des auteurs et de leurs applications. [Kaur,2019].	5
1.2	Schéma d'attribution des auteurs [Roy,2019].	6
1.3	Schéma de vérification des auteurs [Roy,2019].	7
1.4	Schéma de profilage des auteurs [Roy,2019].	8
1.5	La répartition des travaux connexes de Profilage des auteurs.	11
1.6	La répartition des travaux liés au profilage des influenceurs en cryptomonnaie.	15
2.1	Architecture globale de notre approche proposée.	18
2.2	Description de l'ensemble de données.	20
2.3	Préparation des données.	21
2.4	Processus de prétraitement des tweets.	22
2.5	Transformations textuelles avant et après chaque prétraitement.	23
2.6	Architecture de notre approche proposée avec le modèle SetFit.	26
2.7	Architecture du DistilBERT.	31
3.1	Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets burts.	39
3.2	Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets burts.	40
3.3	Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets bruts.	40
3.4	Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets bruts.	41
3.5	Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets bruts.	41
3.6	Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets bruts.	42
3.7	Le graphe du F1-score des trois sous-tâches utilisant SetFit sur des tweets burts.	43
3.8	Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets prétraités.	44
3.9	Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets prétraités.	44

3.10	Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets prétraités.	45
3.11	Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets prétraités.	45
3.12	Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets prétraités.	46
3.13	Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets prétraités.	46
3.14	Le graphe du F1-score des trois sous-tâches utilisant SetFit sur des tweets prétraités.	47
3.15	L'interface au lancement de l'application.	51
3.16	L'interface pour choisir le type de profilage du l'influenceur.	51
3.17	L'interface avec un exemple de profilage du l'influenceur d'un tweet.	52

Liste des tableaux

1.1	Résumé des travaux connexes de cryptomonnaie.	14
2.1	Les détails des techniques.	30
2.2	Les détails des techniques.	34
3.1	Les performances du DistilBERT sur les tweets bruts en utilisant diverses mesures d'évaluation pour les trois sous-tâches.	39
3.2	Les performances du SetFit sur les tweets bruts en utilisant diverses mesures d'évaluation pour les trois sous-tâches.	42
3.3	Les performances du DistilBERT sur les tweets prétraité en utilisant diverses mesures d'évaluation pour les trois sous-tâches.	43
3.4	Les performances du SetFit sur les tweets prétraité en utilisant diverses mesures d'évaluation pour les trois sous-tâches.	47
3.5	Les résultats des deux modèles sont comparés en termes de F1-score, avec et sans prétraitement.	48
3.6	Analyse comparative des résultats par rapport à l'état de l'art en utilisant F1-score.	50

Liste des algorithmes

1	Proposed algorithm	19
---	------------------------------	----

Introduction Générale

Contexte de travail

Les médias sociaux et le contenu généré par les utilisateurs représentent des technologies interactives destinées à partager des informations, des idées et des intérêts au sein de communautés virtuelles. Ces plateformes varient selon les formats tels que les publications textuelles, les images, les commentaires et les réseaux sociaux. Leur expansion mondiale est manifeste : en janvier 2024[Statisia,2023], il y avait 5,35[Statisia,2023] milliards d'internautes dans le monde, ce qui équivaut à 66,2% [Statisia,2023] de la population mondiale, parmi lesquels 5,04[Statisia,2023] milliards, soit 62,3%[Statisia,2023], étaient des utilisateurs de médias sociaux.

Les cryptomonnaies ont profondément transformé le secteur financier en introduisant des transactions décentralisées et sécurisées grâce à la technologie de la blockchain. Cette innovation repose sur un registre distribué, transparent et immuable qui rend pratiquement impossible la fraude. En permettant des transactions rapides à l'échelle mondiale sans recourir aux intermédiaires traditionnels comme les banques, elles réduisent les coûts et accélèrent les délais. La nature décentralisée des cryptomonnaies donne aux utilisateurs un contrôle total sur leurs fonds, offrant une protection renforcée contre les restrictions financières imposées par les autorités centrales. Cette transition vers des systèmes numériques autonomes et accessibles à l'échelle mondiale reflète une société de plus en plus interconnectée et équitable.

Problématique

Cependant, cette avancée a également vu l'émergence d'influenceurs spécialisés jouant un rôle crucial dans des domaines tels que l'éducation du public, l'analyse de marché, la promotion de projets et la gestion de communautés en ligne. Malgré leurs contributions positives, plusieurs défis persistent, notamment la propagation de désinformation pour manipuler le marché, les conflits d'intérêts non divulgués, ainsi que les fraudes et escroque-

ries, entraînant une volatilité accrue des prix et une perte de confiance généralisée. Les médias sociaux amplifient ces problèmes en facilitant la diffusion de désinformation et la manipulation du marché, soulignant ainsi l'importance de mécanismes de vérification et de surveillance rigoureuse.

Objectifs du travail

Pour faire face à ces défis, le profilage des influenceurs de cryptomonnaies est devenu une stratégie cruciale visant :

Mieux comprendre leur rôle et leur impact.

Identifier les comportements préjudiciables comme les fraudes et les manipulations de marché pour protéger les investisseurs.

Renforcer la transparence et la confiance sur le marché.

En consolidant la crédibilité et la stabilité du marché des cryptomonnaies, cette approche vise à protéger les investisseurs contre les risques potentiels et à maintenir une diffusion d'information fiable et équilibrée.

Organisation du mémoire

Ce mémoire est divisée en trois chapitres suivis d'une conclusion générale.

Chapitre 1 : Dans ce chapitre, nous présentons une vue d'ensemble du domaine de l'analyse des auteurs et de ses catégories. Nous soulignons en particulier le domaine du profilage des auteurs, qui est au cœur de notre étude. Nous présentons également les travaux connexes menés dans ce domaine en décrivant leurs différentes approches et méthodes, ainsi qu'une analyse comparative des résultats obtenus.

Chapitre 2 : Dans le deuxième chapitre, nous nous concentrerons sur l'approche que nous proposons, y compris l'ensemble de données utilisé et l'analyse comparative des résultats obtenus. Nous décrirons également les différentes techniques de prétraitement utilisées. En outre, nous fournirons des descriptions des différents modèles utilisés dans notre tâche et de leurs architectures.

Chapitre 3 : Dans le dernier chapitre de notre mémoire, nous présentons la configuration expérimentale, les mesures d'évaluation utilisées pour évaluer la performance de nos modèles, ainsi que l'analyse de leur performance. Nous concluons ce chapitre en présentant notre interface de profilage des auteurs.

Enfin, nous concluons ce mémoire par une conclusion générale mettant en avant les résultats que nous avons obtenus, tout en réfléchissant également à l'avenir de la recherche. Nous discutons des limites de nos modèles et offrons un aperçu des perspectives d'avenir.

Profilage des auteurs

1.1 Introduction

Le traitement du langage naturel (TLN) est un domaine captivant de l'informatique qui fusionne la linguistique et l'intelligence artificielle pour permettre aux ordinateurs de comprendre le langage humain. Avec l'explosion de la communication numérique et la production massive de données textuelles, le NLP est devenu un outil indispensable pour le traitement et l'analyse des données textuelles.

L'analyse d'auteur est une tâche de traitement du langage naturel (TLN) visant à identifier ou caractériser les auteurs de textes écrits. Cette discipline englobe plusieurs sous-catégories, telles que l'attribution des auteurs, la vérification des auteurs et le profilage des auteurs. Notre travail se concentre spécifiquement sur le profilage des auteurs, en mettant l'accent sur les influenceurs de cryptomonnaies.

Pour mieux comprendre le profilage des influenceurs de cryptomonnaies, nous avons examiné un ensemble d'articles sur ce sujet et extrait les résultats les plus significatifs obtenus grâce à différents modèles d'apprentissage.

1.2 Traitement du langage naturel (TLN)

Le traitement du langage naturel (TLN) est une discipline de l'intelligence artificielle dédiée à la conception et à l'implémentation de méthodes informatiques permettant aux machines de comprendre, interpréter et générer le langage humain de manière contextuelle. Elle englobe des techniques telles que l'analyse syntaxique, la reconnaissance d'entités, la traduction automatique, la génération de texte, et d'autres approches visant à faciliter la communication efficace entre l'homme et la machine [Jurafsky et Martin,2009].

1.3 Analyse des auteurs

L'analyse des auteurs est le processus d'examen systématique des caractéristiques linguistiques d'un texte dans le but d'identifier des modèles spécifiques et d'attribuer le texte à un auteur particulier. Cette approche repose sur l'idée que chaque individu a un style d'écriture unique qui peut être décelé à travers diverses caractéristiques linguistiques [Stamatatos,2009].

L'analyse des auteurs fait partie du traitement du langage naturel et de la linguistique médico-légale. Elle implique l'évaluation des documents pour déterminer leur paternité en se basant sur la stylométrie, un domaine linguistique qui analyse le style littéraire à l'aide de statistiques et qui constitue le fondement de l'analyse de la paternité [Abbasi et Chen,2005].

1.4 Les catégories d'analyse des auteurs

L'analyse des auteurs peut être classée en trois catégories selon la figure 1.1 : l'attribution des auteurs, la vérification des auteurs et la caractérisation des auteurs, qui est notre tâche spécifique, notamment le profilage des auteurs.



FIGURE 1.1 – Présentation des catégories de l'analyse des auteurs et de leurs applications. [Kaur,2019].

1.4.1 Attribution des auteurs

L'attribution des auteurs, ou l'identification, consiste à déterminer l'auteur d'un document anonyme contesté parmi un ensemble d'auteurs candidats pour lesquels des échantillons de texte écrit sont disponibles, en analysant les caractéristiques stylistiques de ce document. Dans la littérature, cela est souvent considéré comme un problème

de catégorisation ou de classification de texte [Elmanarelbouanani et Kassou,2014]. Les premières tentatives de mesure du style d'écriture remontent au 19e siècle avec l'étude révolutionnaire de [Mendenhall,1887] sur les pièces de Shakespeare, suivie des études statistiques de [Yule,1939] et [Zipf,1932]. Plus tard, l'enquête approfondie sur l'attribution des "Papiers Fédéralistes" par [Mosteller et Wallace,1963] a été indiscutablement le travail le plus marquant sur l'attribution d'auteur [Stamatatos,2009]. La figure 1.2 illustre un exemple d'attribution des auteurs.

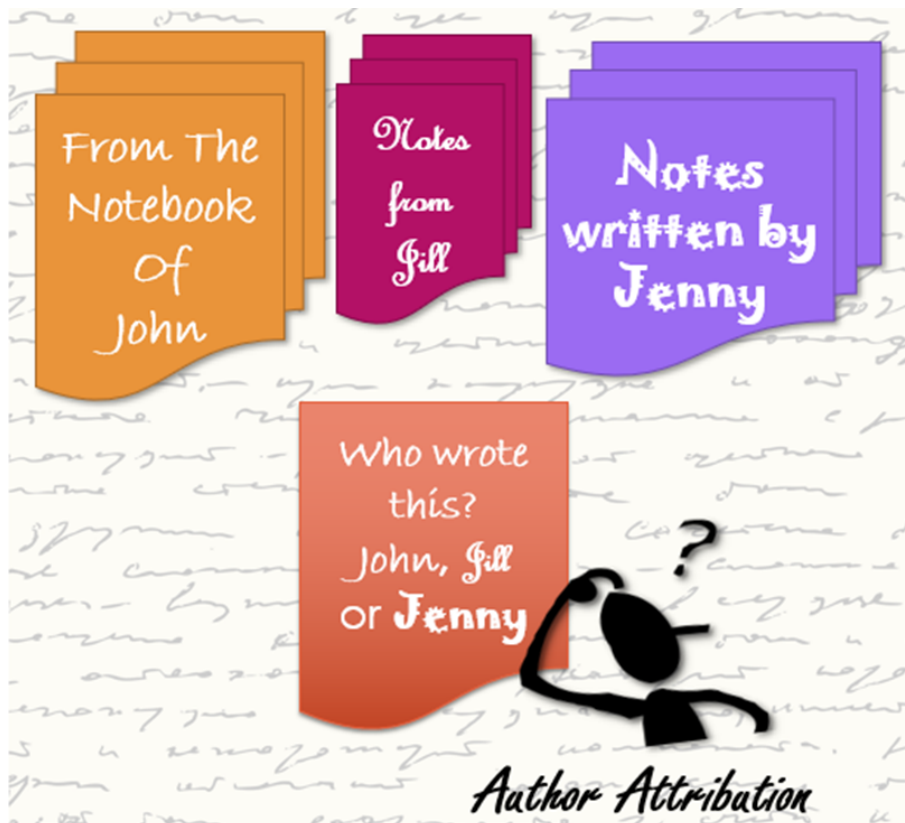


FIGURE 1.2 – Schéma d'attribution des auteurs [Roy,2019].

1.4.2 Vérification des auteurs

La vérification des auteurs d'un texte est le processus qui consiste à déterminer si un texte donné a bien été écrit par un auteur revendiqué ou contesté. Cela implique l'analyse des caractéristiques linguistiques, du style d'écriture et d'autres aspects pour évaluer la cohérence entre le texte contesté et le style d'écriture connu de l'auteur présumé [Juola et al.,2008].

Une autre définition de la vérification des auteurs :

La vérification des auteurs fait référence au processus permettant de déterminer si deux documents pourraient avoir été rédigés par le même auteur [Stamatatos,2016]. Cette tâche représente un défi majeur dans le domaine de l'attribution des auteurs. Contrairement

aux modèles d'attribution d'auteur qui visent à identifier l'auteur le plus probable d'un document anonyme, un modèle de vérification d'auteur doit déterminer si le texte anonyme est suffisamment similaire aux textes connus d'un auteur spécifique pour conclure qu'ils ont été créés par la même personne [Stamatatos,2016]. La figure 1.3 montre un exemple de vérification d'auteur :



FIGURE 1.3 – Schéma de vérification des auteurs [Roy,2019].

1.4.3 Profilage des auteurs

Le profilage des auteurs est une méthode de classification des textes qui vise à détecter des caractéristiques personnelles telles que l'âge, le genre, le niveau d'études, le lieu d'origine, les traits de personnalité, et la langue maternelle des auteurs à partir de leurs écrits. Cette technique trouve des applications diverses dans des domaines comme l'analyse médico-légale, la sécurité, et le marketing, permettant ainsi d'identifier des informations clés sur les auteurs [Para et Patel,2023].

Voici quelques applications du profilage d'auteur dans différents domaines :

En marketing, les systèmes de recommandation jouent un rôle essentiel dans le paysage en ligne actuel. En exploitant les données extraites des profils des clients potentiels, ces systèmes peuvent améliorer la personnalisation des recommandations, offrant ainsi une

meilleure expérience utilisateur. De plus, les grandes entreprises sont souvent intéressées par l'analyse des blogs et des critiques de produits en ligne pour comprendre les préférences des consommateurs à l'égard de leurs produits [Palomino-Garibay et al.,2015].

Dans le domaine médico-légal, le profilage d'auteur s'avère être un outil précieux pour identifier les auteurs de crimes. Que ce soit dans des situations où il y a un grand nombre de suspects à examiner ou un nombre limité, cette approche permet d'analyser les caractéristiques des textes afin de déterminer leur origine. Ainsi, le profilage d'auteur contribue à faciliter les enquêtes et à résoudre les affaires criminelles [Palomino-Garibay et al.,2015].

Lors de la tâche AP partagée de 2019 sur l'ensemble de données PAN, les chercheurs ont utilisé le profilage des auteurs pour distinguer les flux Twitter écrits par des robots ou des humains, et pour établir le profil de genre des auteurs humains [Daelemans et al.,2019].

Dans une étude récente [Das, Raychawdhary, Dozier, et Seals,2022], le profilage des auteurs a été employé pour identifier les utilisateurs de Twitter diffusant de l'ironie et des stéréotypes en anglais.

Ces applications montrent comment le profilage des auteurs peut être une technique puissante pour extraire des informations précieuses à partir de textes écrits, influençant ainsi divers domaines d'application. La figure 1.4 illustre un exemple de profilage d'auteur.

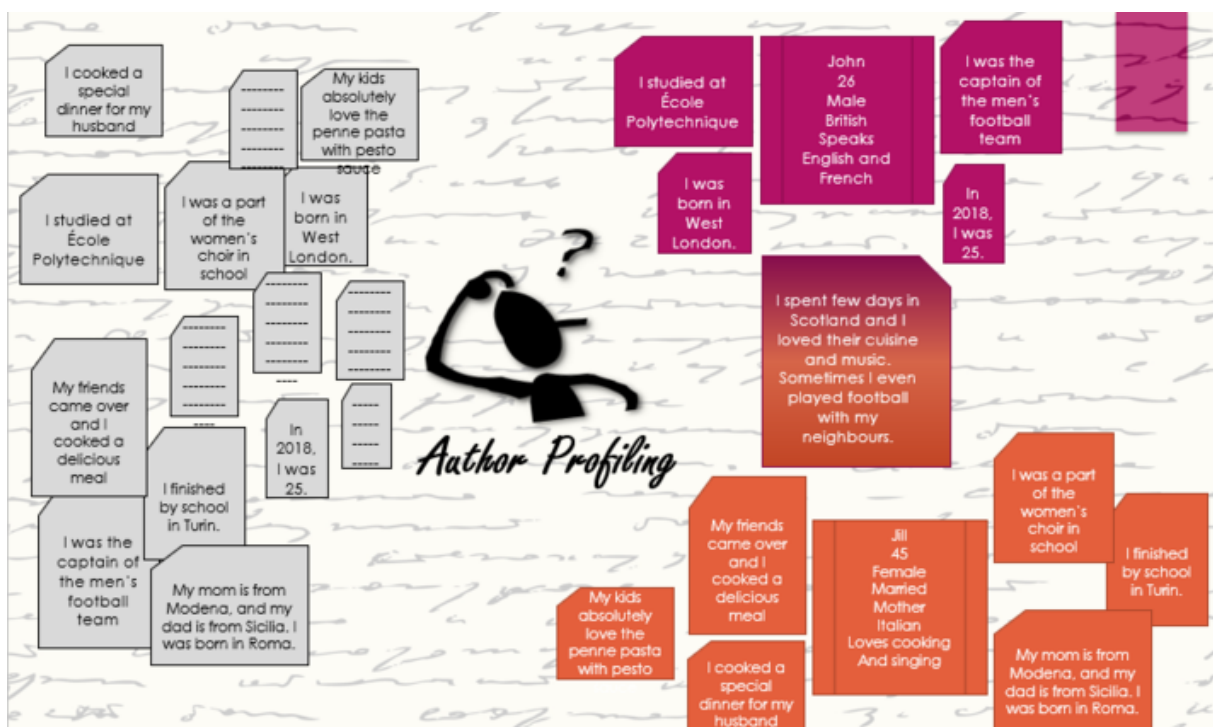


FIGURE 1.4 – Schéma de profilage des auteurs [Roy,2019].

Notre étude se concentrera sur l'approche de profilage des auteurs en ensemble fermé, en ciblant spécifiquement le profilage des influenceurs de cryptomonnaies.

1.5 Travaux connexes

Dans cette section, nous présentons des travaux connexes réalisés dans le domaine de l'analyse des auteurs. Les travaux pertinents peuvent être classés selon la méthode utilisée (par exemple, l'apprentissage automatique ou l'apprentissage profond) et les informations de l'article (telles que le modèle utilisé, l'ensemble de données et ses caractéristiques, ainsi que les résultats obtenus).

1.5.1 Profilage des auteurs

Le profilage des auteurs, une discipline de la linguistique computationnelle, cherche à déterminer l'identité ou les caractéristiques d'un auteur à partir de textes écrits. Son utilisation s'étend à la détection de plagiat, à l'analyse de l'opinion, à la classification des documents et à la résolution d'enquêtes criminelles. En analysant le style d'écriture, le choix du vocabulaire et d'autres traits linguistiques, les chercheurs extraient des informations sur l'auteur d'un texte. Ce résumé met en lumière l'importance de l'identification des auteurs, ses applications, et les méthodes utilisées pour relever ses défis.

L'apprentissage Automatique

Dans [Para et Patel,2023] Développent une nouvelle méthode de représentation des termes pour prédire le genre et l'âge des auteurs, en introduisant une mesure du poids terminologique (TWM). Leur méthode a montré des précisions élevées avec les algorithmes SVM(Support Vector Machine) et la Forêt Aléatoire (RF) sur les ensembles de données PAN2014 et PAN2016.

- Pour PAN2016, ils ont obtenu des précisions de 77,68% (SVM) et 79,42% (RF) pour l'âge, et de 86,23% (SVM) et 88,74% (RF) pour le genre.
- Pour PAN2014, les précisions étaient de 80,71% (SVM) et 82,59% (RF) pour l'âge, et de 85,32% (SVM) et 87,49% (RF) pour le genre.

Dans un contexte différent, [Radha et Chandra Sekhar,2022] se concentrent sur l'analyse des auteurs en utilisant les données des médias sociaux pour les catégoriser par âge, genre et diversité linguistique, avec les données PAN2014. Ils ont employé des modèles d'apprentissage automatique tels que la Forêt Aléatoire, SVM(Support Vector Machine) et NBM(Naive Bayes Multinomial), ainsi que des techniques de sélection de fonctionnalités comme Chi-carré, Information Mutuelle, Gain d'Information et DCC.

Les résultats ont montré une précision de 88% pour le sexe et 81% pour l'âge avec la Forêt Aléatoire, 84,56% pour le genre et 74,91% pour l'âge avec NBM, et 86,12% pour le genre et 78,64% pour l'âge avec SVM, grâce à l'algorithme de sélection de fonctionnalités DCC.

L'apprentissage profond

Une architecture d'apprentissage en profondeur utilisant des modèles BERT et RoBERTA pré-entraînés pour l'espagnol a été décrite dans l'étude de [Carrasco et Rosillo,2022], appliquée à l'ensemble de données PoliticEs 2022. Cette architecture a été efficace pour identifier le genre, la profession et l'idéologie politique à partir des médias sociaux, obtenant une moyenne de F1-score de 90,22% dans la tâche PoliticEs. Les résultats ont démontré sa capacité à extraire automatiquement les caractéristiques des auteurs, contribuant ainsi à l'identification et au ciblage de différents groupes sur les réseaux sociaux.

Dans un autre contexte, [Kowsari et al.,2021] se sont concentrés sur l'application des techniques d'apprentissage profond pour classifier des documents provenant de réseaux sociaux, en se basant sur des données Twitter. Leur approche utilisait un modèle d'apprentissage en profondeur multi-modèles comprenant CNN et DNN.

Les résultats de leur analyse ont montré une précision de 86,33% et un F1-score de 85,83%.

L'apprentissage hybride

[Das et al.,2022] Développé un système pour détecter l'ironie et les stéréotypes dans les tweets en anglais. Leur approche, utilisant un ensemble de données de 600 auteurs avec 200 tweets chacun, combine BERT et TF-IDF pour représenter les tweets. Ils ont trouvé que la régression logistique était l'algorithme le plus performant pour classifier les auteurs propagateurs d'ironie, atteignant une précision de 92,8% sur les données de validation. Pour la détection de l'ironie, le score F1 global était de 45%, avec un score F1 de 19% pour les cas favorables. Sur l'ensemble de test, la précision de la détection de l'ironie avec BERT-TFIDF et la régression logistique était de 67%, tandis que pour la détection des stéréotypes, elle atteignait 58%.

Dans un autre contexte, [Joo et al.,2019] ont participé à la tâche PAN2019, se concentrant sur le profilage des auteurs à partir de tweets en anglais et en espagnol. Leur approche combinait le modèle BERT avec d'autres modèles pour identifier les bots et le genre des auteurs humains. Ils ont obtenu des scores remarquables, avec une précision de 93,33% pour la détection des bots et de 83,52% pour la détermination du genre sur des ensembles de test en anglais.

L'étude met en évidence l'importance de combiner diverses approches pour améliorer le profilage des auteurs.

La figure 1.5 illustre la répartition des travaux liés au profilage d'auteur.

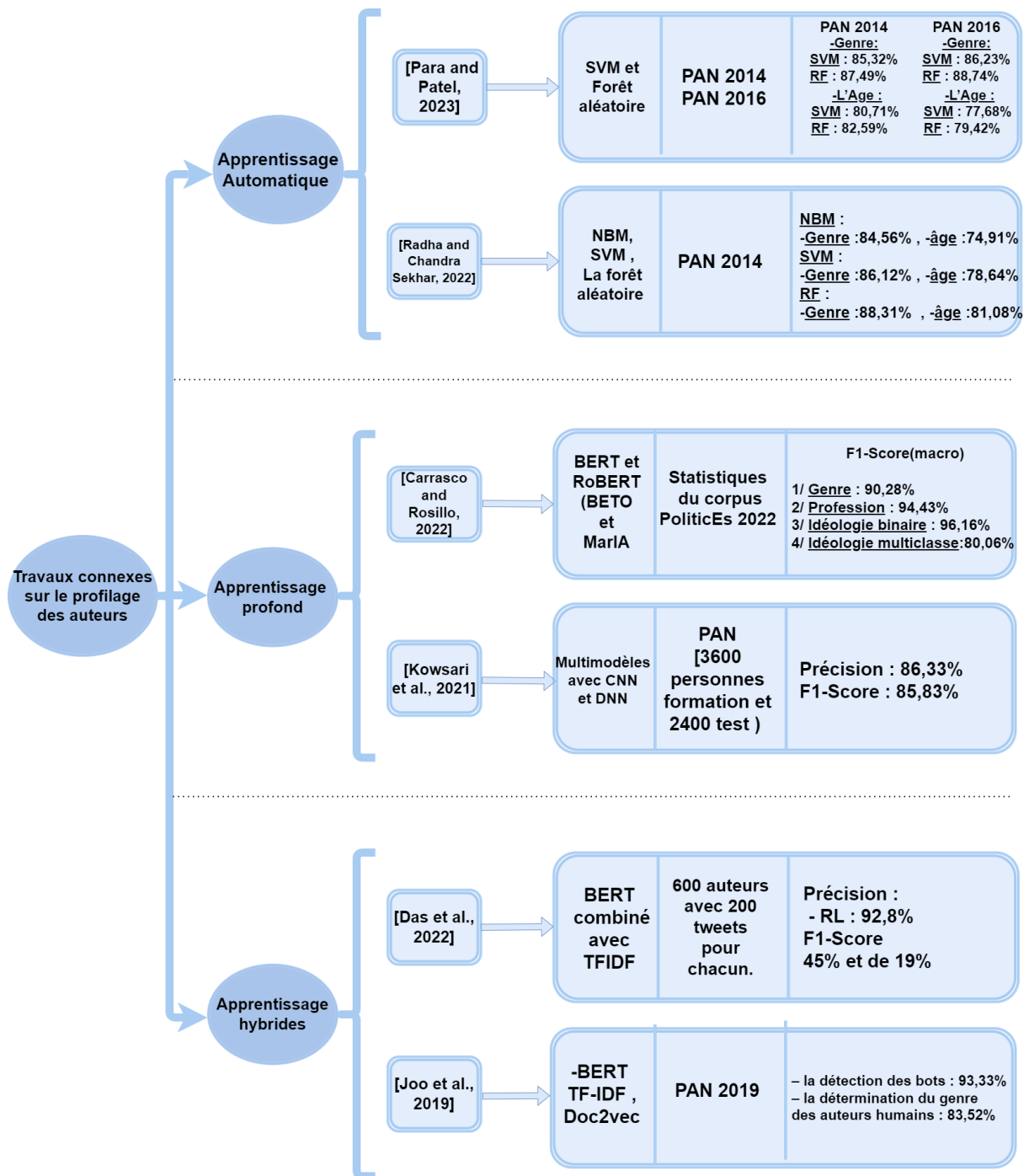


FIGURE 1.5 – La répartition des travaux connexes de Profilage des auteurs.

1.5.2 Cryptomonnaies

Le profilage des influenceurs de cryptomonnaies consiste à examiner et à classer les individus qui influencent les opinions et les décisions dans ce domaine, en prenant en considération leur niveau d'influence, leurs intérêts et leurs intentions. C'est une forme spécifique de profilage d'auteur qui évalue leur impact et leur crédibilité sur les marchés financiers des cryptomonnaies.

Qui joue un rôle essentiel dans une variété de domaines, notamment le marketing, la criminalistique et la sécurité.

Dans cette sous-section, nous avons analysé plusieurs articles pour approfondir notre compréhension du concept de profilage des influenceurs dans le domaine des cryptomonnaies, en extrayant les résultats significatifs obtenus grâce à différentes méthodes d'apprentissage.

Travaux connexes sur le profilage des influenceurs en cryptomonnaies

[Girish, Hegdev, Balouchzahi, et Lakshmaiah,2023] Explorent le domaine de l'apprentissage à quelques échantillons (Few-Shot Learning - FSL). Cette étude se concentre particulièrement sur les défis associés aux ressources limitées. Ils abordent spécifiquement le problème du "Profilage des influenceurs de cryptomonnaie avec l'apprentissage à quelques échantillons", en utilisant deux modèles :

- i) FSL-Word2Vec, un classificateur vectoriel à support linéaire (LinearSVC) entraîné avec des embeddings de mots provenant de Word2Vec pré-entraîné par Google.
- ii) FSL-ST, qui implique l'entraînement d'un LinearSVC avec des embeddings de phrases obtenus à partir de stsb-bert-base.

Les résultats révèlent que le modèle FSL-Word2Vec a obtenu des scores F1-score (macro) de 46,66% et 50,42% pour les sous-tâches 2 et 3 respectivement, tandis que le modèle FSL-ST a enregistré un F1-score (macro) de 37,92% pour la sous-tâche 1.

[Siino, Tesconi, Tinnirello, et al.,2023] Proposent d'élargir la base de données en utilisant la rétrotraduction. Ils ont constaté que les résultats obtenus avec la version italienne (94,31%) et la version allemande (93,11%) étaient supérieurs à ceux de la base non augmentée (37,62%). Ces résultats montrent que l'augmentation des échantillons avec plusieurs langues grâce à la rétrotraduction permet d'améliorer les performances dans les tâches de profilage des auteurs.

[Espinosa et Sidorov,2023] Abordent le défi de repérer les influenceurs des cryptomonnaies sur les réseaux sociaux à partir de l'ensemble de données PAN2023. Ils proposent une solution utilisant les modèles BERT, BERTweet et RoBERTa avec une couche de prétraitement. Les résultats montrent une précision de plus de 92% pour BERT, de 86% pour BERTweet et plus de 93% pour RoBERTa lors du défi PAN2023. Bien que RoBERTa offre des améliorations par rapport à BERT, les auteurs notent ses exigences en temps et en ressources plus élevées, ce qui les conduit à ne pas privilégier RoBERTa, ses résultats n'étant pas jugés significativement supérieurs à ceux de BERT. Ils soulignent l'importance d'identifier des experts ou influenceurs dans le domaine des cryptomonnaies, tout en mettant en avant la complexité du processus de profilage de ces influenceurs par un apprentissage en quelques étapes (Few-Shot Learning - FSL).

[Siino et Tinnirello,2023] Présentent l'application de XLNet pour le profilage des influenceurs de cryptomonnaie à PAN@CLEF2023. Dans un ensemble de 32 utilisateurs (avec 10 tweets en anglais pour chaque auteur de l'ensemble de formation), en utilisant un processus d'apprentissage en quelques étapes (Few-Shot Learning - FSL). En ajustant XLNet sur un ensemble de données augmenté via la rétro-translation en allemand et en italien, notre approche montre des résultats significativement améliorés. Sur l'ensemble non augmenté, les performances pour l'allemand sont un F1-score (macro) de 56,88% avec une précision de 59,37%, et pour l'italien, un F1-score (macro) de 69,37% avec une précision de 68,93%. Sur l'ensemble augmenté, la version peaufinée de XLNet avec l'italien surpasse l'allemand avec une marge de plus de 10% en termes de F1-score (macro) et environ 9% en termes de précision.

[Muslihuddeen, Sathvika, Sankar, Ostwal, et Kumar,2023] Proposent une approche pour classifier les influenceurs en fonction de leur niveau d'influence en utilisant les tweets en anglais du défi PAN2023. Les auteurs ont utilisé des algorithmes tels que SVM, forêt aléatoire et régression logistique, chacun combiné avec l'apprentissage actif, ainsi que la régression logistique avec apprentissage par transfert. Les résultats ont montré que le modèle de régression logistique avec apprentissage actif offrait les meilleures performances, avec un taux de réussite de 34%.

[Ferri-Molla et Santamaria-Jorda,2023] Abordent le défi du profilage des influenceurs de cryptomonnaies à partir de données limitées obtenues sur les réseaux sociaux, en se basant sur les messages en anglais sur Twitter dans l'ensemble de données PAN2023, couvrant les trois sous-tâches. Ils ont utilisé diverses méthodes statistiques telles que K-means, régression logistique, perceptron, SVM, KNeighbors, MLP (perceptron multicouche), Naive Bayes, forêt aléatoire, Ridge, SGD, SVC, ainsi que des modèles de langage pré-entraînés spécifiques comme BERT-base-uncased, Twitter-roberta-base, BERTweet-base et DistilBERT fine-tuned.

Cette approche visait à comparer les performances des modèles statistiques et des modèles de langage pré-entraînés pour chaque tâche. Les résultats expérimentaux ont montré que les modèles pré-entraînés ont généralement obtenu de meilleures performances globales, même avec de faibles quantités de données disponibles.

Le tableau 1.1 ci-dessous résume les recherches antérieures sur les cryptomonnaies en fonction du modèle utilisé, des ensembles de données utilisés et des résultats obtenus pour chaque article.

Article	Modèle	Ensemble de données	Résultats
[Girish et al.,2023]	FSL-Word2Vec FSL-ST	PAN2023 (Sous-tâche1 :160 utilisateurs[5 étiquettes] Sous-tâche2 :320 utilisateurs[4 étiquettes] Sous-tâche3 :256 utilisateurs[4 étiquettes])	F1-score(macro) : -Sous-tâche1= 37,92% avec le modèle FSL-ST -Sous-tâche2 = 46,66% & -Sous-tâche3 = 50,42% avec FSL-Word2Vec
[Siino et al.,2023]	Rétro-traduction (back-translation) pour entraîner ELECTRA	PAN2023	Médiane F1-score(macro) Avec augmentation - Italiennes : 94,31% - Allemands : 93,11% . Sans augmentation = 37,62%
[Espinosa et Sidorov,2023]	BERT BERTweet DistilBERT	PAN2023	F1-score — BERT : — ST1 : 86.77%, ST2 : 83.44%, ST3 : 85.01% Précision — BERTweet : — ST1 : 86,98%, ST2 : 88,76%, ST3 : 92,33% - DistilBERT : - ST1 : 83.29%, ST2 : 85.12%, ST3 : 85,98%
[Siino et Tinirello,2023]	XLNet	32 utilisateurs (10 tweets en anglais pour chaque utilisateur).	F1-score(macro) : Données d'origine - Allemand :56,88% - Italien :69,37% Données augmentées -l'italien surpasse l'allemand avec une marge de plus de 10%
[Muslihuddeen et al.,2023]	-Forêt aléatoire avec et sans apprentissage actif -Régression logistique avec et sans apprentissage actif -SVM avec et sans apprentissage actif -Régression logistique avec apprentissage par transfert	PAN2023(sous-tâche 1 : 160 utilisateur avec 5 classe)	SVM : Précision : 12%, F1-score(macro) :5% Régression logistique : précision : 29%, F1-score(macro) : 26% Forêt aléatoire : précision : 26%, F1-score(macro) : 23% Forêt aléatoire & apprentissage actif : précision : 29%, F1-score(macro) : 24% SVM & apprentissage actif : précision : 14%, F1-score(macro) : 8% Régression logistique & apprentissage actif : précision : 34%, F1-score(macro) : 32% Régression logistique & apprentissage par transfert : précision : 34%, F1-score(macro) : 23%
[Ferri-Molla et Santamaria-Jorda,2023]	- K-means, - Régression logistique, - Perceptron, - SVM, - KNeighbors - MLP(Multilayer Perceptron), - Naive Bayes, - Perceptron - Forêt aléatoire, - Ridge, - SGD, - SVC - bert-base-uncased, - twitter-roberta-base - bertweet-base, - DistilBERT fine-tuned	PAN2023 (Sous-tâche1 :160 utilisateurs[5 étiquettes] Sous-tâche2 :320 utilisateurs[4 étiquettes] Sous-tâche3 :256 utilisateurs[4 étiquettes])	(Précision , F1-score(macro)) SOUS-TÂCHE1 : K-means : (25% , 12%) Régression logistique : (56% , 57%) Perceptron : (51% , 52%), SVM : (57% , 57%) bert-base-uncased : (61% , 59%), twitter-roberta-base : (57% , 57%) bertweet-base : (56% , 55%) SOUS-TÂCHE2 : KNeighbors : (35% , 36%) Régression logistique : (42% , 43%) MLP : (40% , 40%), Perceptron : (41% , 39%) Forêt Aléatoire : (47% , 46%) Ridge : (45% , 46%), SGD : (43% , 41%) SVC : (44% , 43%), SVM : (43% , 42%) DistilBERT fine-tuned : (63%, 63%) twitter-roberta-base : (63%,57%) SOUS-TÂCHE3 : SVM : (62%, 61%) Perceptron : (59%, 58%) Régression logistique : (56%, 58%) DistilBERT fine-tuned : (83%, 84%)

TABLE 1.1 – Résumé des travaux connexes de cryptomonnaie.

La figure 1.6 montre la répartition des travaux liés au profilage des influenceurs de la cryptomonnaie.

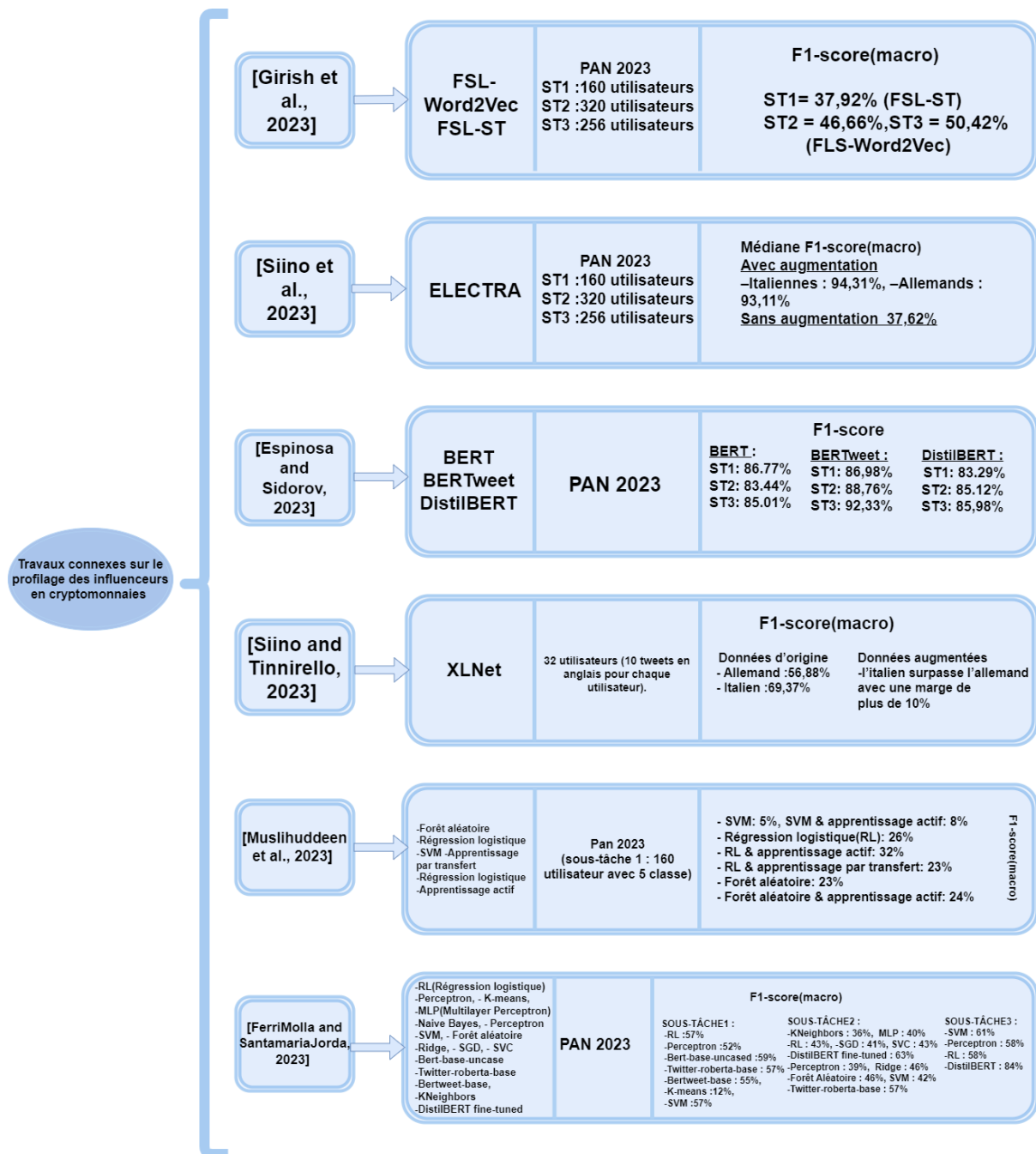


FIGURE 1.6 – La répartition des travaux liés au profilage des influenceurs en cryptomonnaie.

1.6 Discussion

Le tableau 1.1 ci-dessus synthétise les recherches connexes sur le profilage des influenceurs de cryptomonnaies, mettant en lumière l'impact et les résultats des différentes méthodes utilisées. Les études utilisant des techniques d'apprentissage par transfert avec

un faible nombre d'exemples se démarquent par leurs performances remarquables. Ces méthodes d'apprentissage par transfert ont atteint une précision supérieure à 85%, tandis que les approches traditionnelles ont rarement dépassé les 35%. Parmi ces dernières, on compte le SVM, la forêt aléatoire et la régression logistique.

En revanche, parmi les méthodes d'apprentissage par transfert les plus performantes, on retrouve BERT, DistilBERT et ELECTRA, entre autres. Ces techniques ont été appliquées avec succès au jeu de données PAN2023, qui comprend trois sous-tâches distinctes :

- **Profilage des influenceurs selon leur degré d'influence** avec des ressources limitées : Identifier et caractériser les influenceurs en utilisant un nombre restreint de données disponibles..
- **Identification des intérêts des influenceurs** avec des ressources limitées : Déterminer les centres d'intérêt des influenceurs en utilisant un nombre limité d'exemples.
- **Identification de l'intention des influenceurs** avec des ressources limitées : Discerner les intentions derrière les actions et les messages des influenceurs en utilisant également des ressources limitées..

Ces résultats soulignent l'efficacité des techniques d'apprentissage par transfert dans le domaine du profilage des influenceurs de cryptomonnaies, dépassant de loin les méthodes traditionnelles en termes de précision et de robustesse.

1.7 Conclusion

Dans ce chapitre, nous avons exploré le domaine de l'analyse des auteurs et ses différentes catégories, à savoir l'attribution des auteurs, la vérification des auteurs et la caractérisation des auteurs. Une attention particulière a été accordée à la caractérisation des auteurs, notamment au profilage des auteurs, en nous concentrant spécifiquement sur le cas du profilage des influenceurs de cryptomonnaie, qui constitue notre sujet d'étude. Nous avons également passé en revue divers travaux connexes et les méthodes utilisées pour aborder le problème du profilage des influenceurs de cryptomonnaie sur les réseaux sociaux. Après avoir examiné l'ensemble des articles étudiés, il apparaît que les meilleurs résultats ont été obtenus grâce à des méthodes d'apprentissage par transfert.

Dans le chapitre suivant, nous présenterons notre approche ainsi que les méthodes que nous avons utilisées.

Solution proposée

2.1 Introduction

Traditionnellement, la caractérisation des influenceurs dans le domaine des cryptomonnaies repose sur la connaissance de leur niveau d'influence dans ce secteur ainsi que sur l'identification de leurs intérêts et intentions. Les avancées en matière d'apprentissage automatique, d'apprentissage profond et d'apprentissage par transfert ont introduit des techniques plus sophistiquées pour déterminer ces caractéristiques. Des travaux antérieurs ont démontré que l'utilisation de ces approches peut considérablement améliorer les résultats de cette tâche.

Dans ce chapitre, nous présentons notre architecture proposée pour résoudre le problème de l'identification des influenceurs dans le domaine des cryptomonnaies. Nous décrivons l'ensemble de données utilisé ainsi que les étapes nécessaires à la préparation et au prétraitement de nos données.

2.2 Architecture globale

Pour résoudre le problème de l'identification des influenceurs dans le domaine des cryptomonnaies, nous avons mis en œuvre une architecture illustrée dans la figure 2.1, suivie d'une explication détaillée de ces étapes.

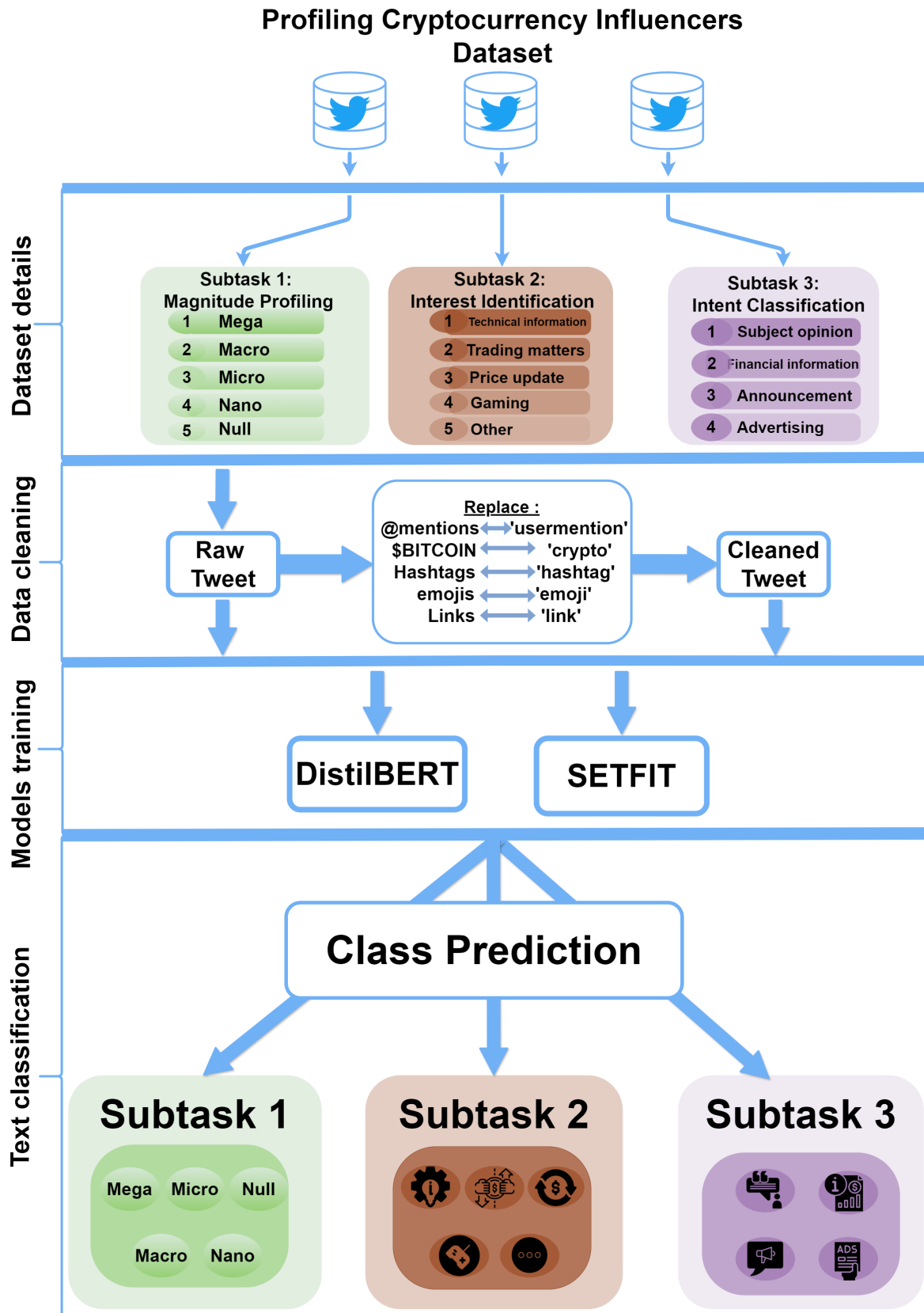


FIGURE 2.1 – Architecture globale de notre approche proposée.

Prétraitement des données : Cette étape implique le nettoyage, la normalisation et la structuration du texte brut non structuré. Cela inclut la suppression des données superflues telles que les identifiants d'utilisateur et d'article, ainsi que l'application de techniques de prétraitement pour assurer la cohérence et la lisibilité du texte.

Division des données : Cette étape consiste à séparer les données en deux ensembles distincts : l'ensemble d'entraînement et l'ensemble de test.

Sélection du modèle : À ce stade, nous avons choisi deux modèles pré-entraînés spécialisés dans la classification de textes parmi les architectures d'apprentissage par transfert disponibles.

Évaluation du modèle : Enfin, nous avons évalué les performances de nos modèles en utilisant de nouvelles données (ensemble de test) afin de mesurer leur efficacité. Voici un pseudo code proposé pour notre approche :

Algorithm 1 Proposed algorithm

```
1: Input : dataset
2: Output : evaluation mesures (Accuracy, F1-score)
3:   For each tweet in dataset
4:   tweet ← replace ({('https:', 'link'), ('$...', 'crypto'), ('@', 'usermention'), ('#', 'hash-tag'), ('Emoji representation', 'emoji')} in tweet)
5:   End For
6:   train_data, test_data ← split(dataset)
7:   If model is "Paraphrase-mpnet-base-v2"
8:   positive triplet, négative triplet ← (Generate pairs of tweets along with their corresponding class.)
9:   new dataset ← concatenate(positive triplet, négative triplet)
10:  model ← Fine_tuned Paraphrase-mpnet-base-v2 (new dataset)
11:  For each tweet in test_data
12:  Embedding ← Fine_tuned Paraphrase-mpnet-base-v2 (tweet)
13:  Prédiction class ← Classification head (Embedding)
14:  EndFor
15:  Evaluate Accuracy, F1-score
16:  Else model is "Distil BERT"
17:  Token ← Tokenisation(train_data)
18:  Token ← Tokenisation(test_data)
19:  model ← Fine_tuned Distil BERT (train_data)
20:  For each tweet in test_data
21:  The latent représentation ← Fine_tuned Distil BERT(tweet)
22:  Prédiction class ← Classification head (The latent représentation)
23:  End For
24:  Evaluate Accuracy, F1-score
25:  End If
26:  return(Accuracy, F1-score)
```

2.3 L'ensemble de données

Depuis 2010, PAN¹ organise une série d'événements scientifiques et de tâches collaboratives dédiés à la criminalistique des textes numériques et à la stylométrie. L'édition PAN @CLEF 2023 "Profiling Cryptocurrency Influencers with Few-shot Learning"[Chinea-Rios, Borrego-Obrador, Franco-Salvador, Rangel, et Rosso,2023] est une tâche partagée qui se concentre sur le profilage des influenceurs de cryptomonnaies sur les médias sociaux avec des ressources limitées [given i=F, given i=M, given i=M, et given i=P,s. d.]. Cette initiative se focalise sur l'analyse des messages Twitter en anglais et propose trois sous-tâches distinctes, dont les détails sont présentés dans la figure 2.2. L'objectif est de surmonter les défis liés à la classification de texte dans un contexte où les ressources sont restreintes.

Tâches	Nom tâche	Classes	Description de la Distribution du label	Ensemble d'entraînement par utilisateurs	Ensemble de test par utilisateurs
Sous-tâche 1	Profilage des influenceurs à faibles ressources (selon le type d'influenceur de l'auteur)	(1) nul	32 utilisateurs par étiquette avec un maximum de 10 tweets en anglais chacun 929 tweets	128	32
		(2) nano			
		(3) micro			
		(4) macro			
		(5) méga			
Sous-tâche 2	Identification des intérêts des influenceurs disposant de peu de ressources	(1) informations techniques	64 utilisateurs par étiquette avec 1 tweets en anglais 320 tweets	256	64
		(2) mise à jour des prix			
		(3) questions commerciales			
		(4) jeux			
		(5) autres			
Sous-tâche 3	Identification des intentions des influenceurs à faibles ressources	(1) opinion subjective	64 utilisateurs par étiquette avec 1 tweets en anglais chacun 256 tweets	204	51
		(2) information financière			
		(3) publicité			
		(4) annonce			

FIGURE 2.2 – Description de l'ensemble de données.

2.4 Préparation des données

Notre jeu de données se compose de trois fichiers, chacun contenant des données au format JSON :

- Le premier fichier, nommé "*train-text.json*", comprend trois colonnes : "*twitter user id*", "*texts*", et "*tweets ids*".
- Le deuxième fichier, nommé "*train-truth.json*", contient deux colonnes : "*twitter user id*" et "*class*", comme illustré dans la figure 2.3 ci-dessous."

1. <https://pan.webis.de/>

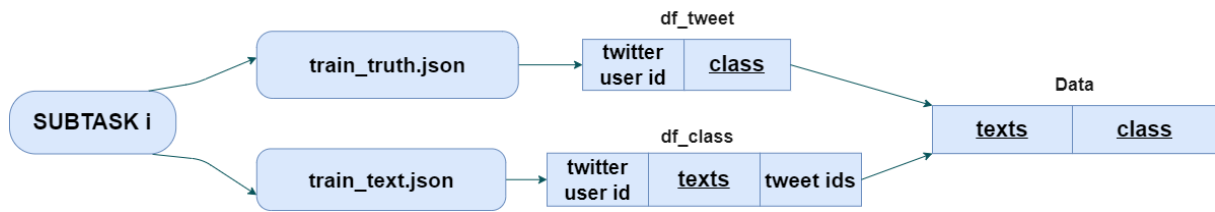


FIGURE 2.3 – Préparation des données.

Dans notre approche, nous avons désigné :

- Le fichier "*train-text.json*" comme "*df-tweet*".
- Le fichier "*train-truth.json*" comme "*df-classe*".
- La colonne "*texts*" dans "*df-tweet*" contient les tweets des utilisateurs. Chaque tweet est précédé du mot "*text*". Nous avons retiré ce mot de chaque tweet et fusionné le reste en utilisant un espace comme séparateur pour chaque ligne.
- Ensuite, nous avons fusionné les deux *DataFrames*, *df-tweet* et *df-classe*, selon la colonne "*twitter user id*" dans le *DataFrame* "*Data*" en utilisant la ligne de code suivante :
- Après cela, nous avons supprimé les colonnes inutiles telles que "*tweet ids*" et "*twitter user id*" de "*Data*".
- Enfin, nous avons créé un nouveau *DataFrame* nommé "*df-data*", contenant les tweets avec leur classe encodée.

L'objectif est de s'assurer que les données d'entrée sont correctement structurées et prêtes à être utilisées par le modèle pour le prétraitement.

2.5 Prétraitement

Dans le cadre de toute tâche impliquant du texte, il est essentiel d'appliquer systématiquement une couche de prétraitement avant d'utiliser directement les modèles. Le prétraitement des données consiste à transformer, nettoyer et préparer ces dernières en vue de l'analyse ou de la modélisation. Cette étape cruciale contribue à améliorer la qualité des données et facilite leur intégration dans les algorithmes et les modèles. Nous avons utilisé la configuration suivante pour les trois sous-tâches, comme illustré dans la figure 2.4.

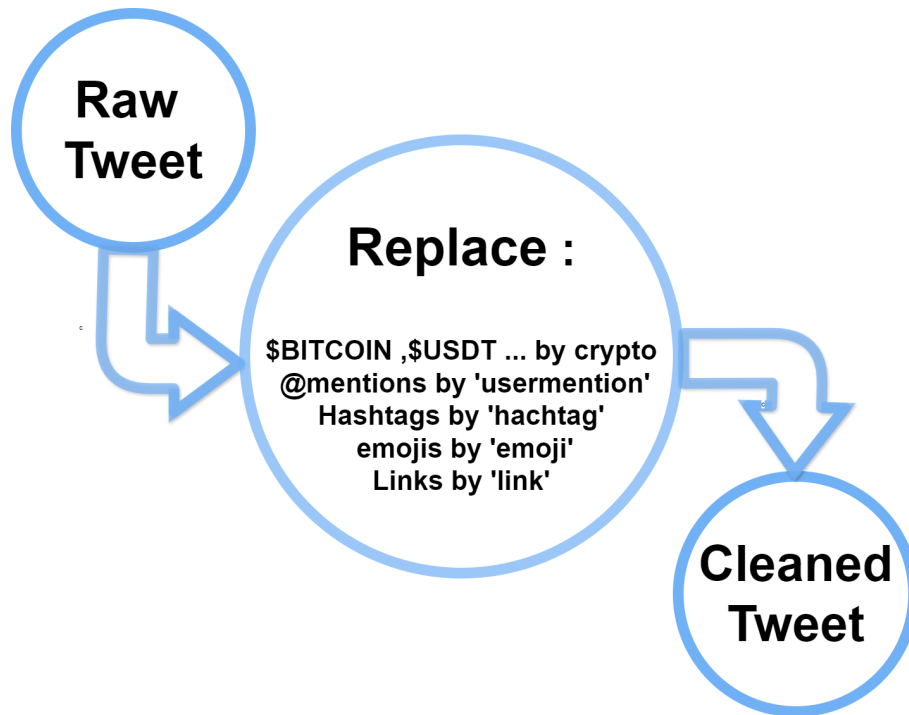


FIGURE 2.4 – Processus de prétraitement des tweets.

Les liens : Remplacer les URLs comme `https://...` par l'étiquette "lien" dans les tweets.

Les cryptomonnaies : Remplacer les noms des cryptomonnaies comme "\$BITCOIN" par l'étiquette 'crypto' dans les tweets.

Mentions : Les mentions, indiquées par le symbole "@", font référence à d'autres utilisateurs sur Twitter.

Remplacer ces mentions par le terme "usermention"

Hashtags : Les hashtags servent à indexer des mots-clés ou des sujets sur Twitter, marqués par le symbole "#".

Remplacer ces hashtags par l'étiquette "hashtag".

Emojis : Les emojis ont été remplacés par le mot "emoji" dans les tweets, une action visant à réduire le bruit dans les tweets.

Remplacer ces emojis par l'étiquette "emoji".

La figure 2.5 illustre les transformations appliquées aux textes avant et après chaque étape de prétraitement mentionnée précédemment.

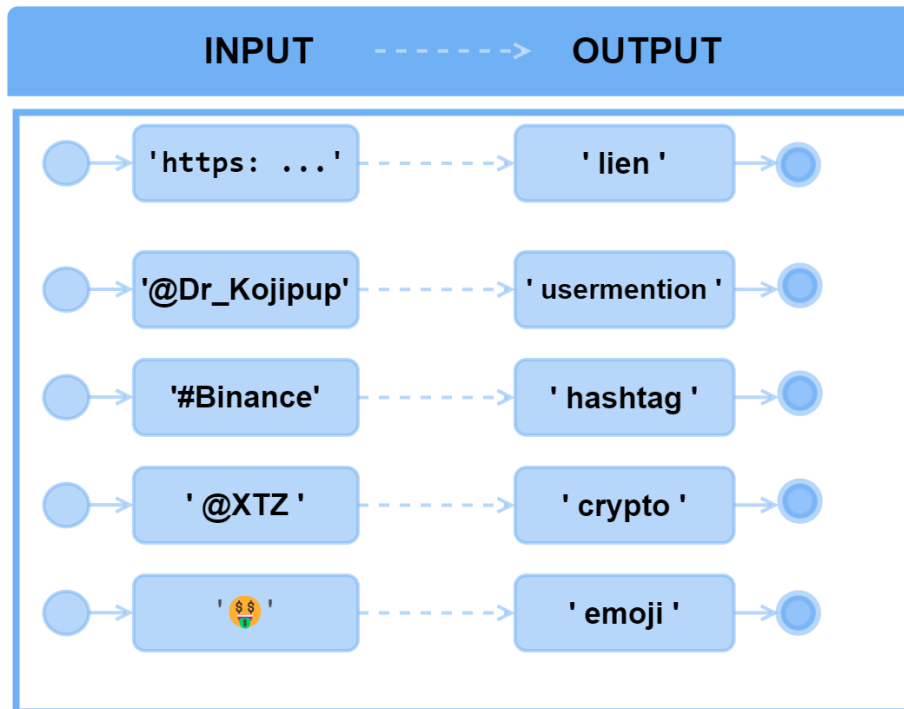


FIGURE 2.5 – Transformations textuelles avant et après chaque prétraitement.

2.6 Ensemble Entraînement/Test

Dans cette étape, nous divisons nos données en deux sous-ensembles pour évaluer de manière impartiale les performances de nos modèles, chacun utilisé pour des tâches spécifiques décrites ci-dessous :

Ensemble d’entraînement : Il représente la plus grande partie de notre ensemble de données et est utilisé pour entraîner notre modèle en lui fournissant les données d’entrée avec les étiquettes correspondantes. Son rôle est de permettre au modèle d’apprendre à partir des données afin de faire des prédictions précises sur de nouvelles données.

Ensemble de test : Il représente une petite partie de l’ensemble de données et est réservé à l’évaluation des performances de nos modèles sur de nouvelles données. Cela permet de tester la capacité du modèle à généraliser à des données qui n’ont pas été utilisées lors de l’entraînement.

Pour notre tâche spécifique, nous avons assigné 80% des données à l’ensemble d’entraînement et 20% à l’ensemble de test.

2.7 Construction des modèles

Dans cette section, nous présentons les modèles que nous avons utilisés pour traiter le problème de profilage des influenceurs de cryptomonnaie en utilisant une approche d'apprentissage en quelques étapes. Cette méthode implique de former un modèle avec un nombre limité d'exemples, également connue sous le nom d'apprentissage en quelques coups ou "few-shot learning". Cette approche permet au modèle d'apprendre efficacement même avec des données d'entraînement limitées, ce qui est particulièrement utile lorsque les données sont rares ou difficiles à obtenir. Cette tâche a été partagée lors de PAN @CLEF 2023, où les chercheurs ont exploré divers modèles transformers tels que Electra, XLNet, et d'autres encore. Dans notre étude, nous proposons d'utiliser deux architectures distinctes : Sentence Transformer Fine-tuning (SetFit) et Distilled Intermediate Smaller Transformer Increased Learning Bidirectional Encoder Representations from Transformers (DistilBERT).

Les deux modèles que nous avons utilisés pour résoudre le défi de l'identification des influenceurs dans le domaine des cryptomonnaies sont tous les deux basés sur l'architecture Transformer.

2.8 Les modèles Transformer

Les modèles Transformer sont un type moderne de réseau neuronal, introduit dans l'article "Attention is All You Need" [Vaswani et al.,2017], qui a révolutionné le traitement automatique du langage naturel (TALN) en utilisant des mécanismes d'auto-attention pour se concentrer sur le texte en entrée. Les modèles Transformer sont utilisés pour résoudre toutes sortes de tâches en TALN, y compris la classification de phrases entières, le résumé de texte, la classification de chaque mot dans une phrase, la génération de contenu textuel et l'extraction de réponses à partir d'un texte. De plus, ces modèles constituent la base de technologies telles que BERT et GPT-4 [Vaswani et al.,2017].

Les modèles Transformer se composent de deux composants principaux : l'encodeur et le décodeur. L'encodeur commence par une couche d'incorporation pour convertir les tokens d'entrée en représentations numériques. Ensuite, il utilise plusieurs couches de transformers pour capter l'attention et comprendre le contexte du texte en entrée. D'autre part, le décodeur génère un texte de sortie en fonction des informations contextuelles apprises par l'encodeur.

Les caractéristiques spécifiques de chaque architecture des deux modèles seront détaillées dans les sections suivantes.

2.8.1 SetFit

L'algorithme SetFit (Sentence Transformer Fine-tuning) est un algorithme récemment introduit par la société Hugging Face². Qui est connue pour ses avancées dans le domaine de l'intelligence artificielle appliquée au traitement du langage naturel fondée en 2016, en partenariat avec d'autres entreprises Intel Labs³ et UKP Lab⁴. L'article de recherche présentant cet algorithme a été publié en septembre 2022 [Tunstall et al.,2022]. Il s'agit d'une méthode d'apprentissage automatique qui utilise un transformateur de phrase pré-entraîné, qui sera ensuite adapté à un problème sélectionné en utilisant un nombre limité de données entrantes. Ce type d'approche dans la littérature est nommé l'apprentissage en quelques coups (Few-shot learning). Cette approche a été popularisée par GPT-3 (Generative Pre-Trained Transformer 3), plus précisément l'outil ChatGPT⁵ développé par OpenAI [Brown et al.,2020]. Bien que l'objectif de ChatGPT soit de créer un robot de conversation (Chat bot) et que l'objectif visé par ce mémoire est de faire de la classification des tweets, la même approche peut être utilisée.

SetFit est basé sur les transformateurs de phrase décrits par [Reimers et Gurevych,2019], qui sont pré-entraînés sur une base de données. Ces transformateurs de phrases génèrent des vecteurs de sortie très denses, utilisables ensuite par des algorithmes d'apprentissage automatique [Bernier,2023]. Une fois un transformateur de phrase sélectionné, l'algorithme SetFit propose de suivre les étapes présentées dans la figure 2.6, après la création d'un ensemble de données d'entraînement(les tweets).

[Tunstall et al.,2022] Les auteurs de cet article ont été les premiers à présenter et introduire, en 2022, le modèle SetFit comme une méthode innovante pour l'apprentissage avec peu d'exemples (few-shot learning). Ils décrivent également en détail les étapes nécessaires pour appliquer SetFit à la classification de texte, qui a été la première utilisation de ce modèle.

[Reimers et Gurevych,2019] Présentent l'architecture de le modèle (SBERT),une version modifiée du modèle pré-entraîné BERT. SBERT utilise des architectures de réseau siamois et triplet pour produire des embeddings de phrases avec une signification sémantique, ce qui facilite leur comparaison à travers la similarité cosinus.

2. <https://huggingface.co/>

3. <https://www.intel.fr/content/www/fr/fr/homepage.html>

4. <http://www.ukp.tu-darmstadt.de/>

5. <https://chatgpt.com/>

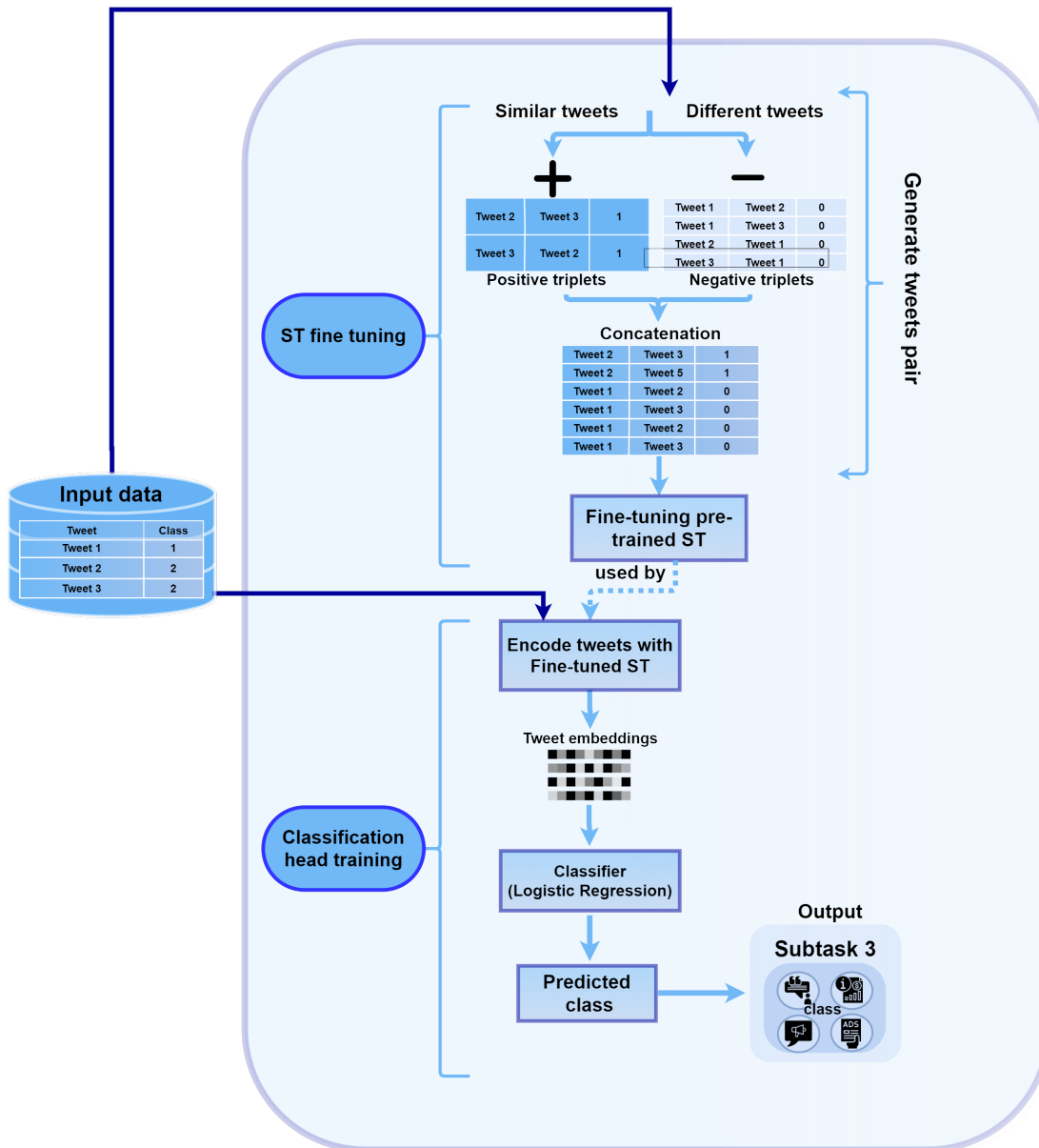


FIGURE 2.6 – Architecture de notre approche proposée avec le modèle SetFit.

Tout d’abord, nous avons chargé le modèle "paraphrase-mpnet-base-v2". Ce modèle a automatiquement effectué plusieurs étapes, notamment le prétraitement et la tokenisation de tous les tweets, ainsi que l’encodage de chaque jeton en vecteur d’embedding. Ensuite, le modèle a réalisé deux étapes, pour lesquelles nous avons utilisé la sous-tâche 3 de notre ensemble de données comme exemple.

La première étape consiste à affiner le modèle pré-entraîné Sentence Transformer (ST). Le modèle génère des paires de tweets avec leurs classes correspondantes, notées

$D=(X, Y)$, où X représente l'ensemble des tweets et Y l'ensemble de leurs classes parmi les quatre catégories suivantes : informations financières, opinion subjective, publicité et annonce. Le modèle crée des triplets positifs $T_{pc}=(x_i, x_j, 1)$ pour des tweets de la même classe et des triplets négatifs $T_{cn}=(x_i, x_j, 0)$ pour des tweets de classes différentes. Il produit un ensemble d'entraînement T en regroupant les triplets positifs et négatifs des étiquettes de classe. L'affinement du modèle pour la tâche spécifique de classification des tweets consiste à ajuster minutieusement ses poids afin d'optimiser sa capacité à distinguer et classer avec précision les tweets selon leurs catégories.

Dans **la deuxième étape**, Classification Head Training, le modèle affiné génère des embeddings pour chaque tweet, capturant ainsi leur représentation sémantique. Ces embeddings sont utilisés comme entrée pour une tête de classification, qui est généralement une régression logistique dans notre modèle par défaut. Cette tête de classification est entraînée à associer les embeddings appris aux étiquettes de classe correspondantes. Lorsqu'un nouveau tweet est introduit, le modèle affiné génère un vecteur d'embedding qui capture l'essence sémantique du tweet. Ce vecteur est ensuite utilisé par la tête de classification pour prédire l'étiquette de classe appropriée. Cette approche garantit une classification précise des textes basée sur leur contenu sémantique. Nous détaillons dans ce qui suit, chacune de ces étapes :

2.8.1.1 Transformateur de phrase (Sentence-transformers)

Dans notre approche, nous utilisons le modèle "sentence-transformers/paraphrase-mpnet-base-v2"

Paraphrase-mpnet-base-v2 : un modèle contenant le plus de dimensions vectorielles comparé aux autres modèles testés. En effet, ce modèle utilise un espace vectoriel dense de 768 dimensions et peut être utilisé pour des tâches telles que le regroupement ou la recherche sémantique[Sentence Transformers,2024].

Dans notre cas, pour la classification, le modèle "paraphrase-mpnet-base-v2" est appliqué pour extraire des caractéristiques significatives des tweets, permettant ainsi de les classer efficacement en différentes catégories ou classes. Cette approche est documentée dans plusieurs recherches et projets dans le domaine du traitement automatique du langage naturel.

2.8.1.2 Affiner le transformateur de phrases pré-entraîné (Sentence Transformer Fine-Tuning)

Cette étape se concentre sur la préparation du modèle pour la représentation des tweets efficace,et elle se déroule comme suit :

1. Premièrement nous **chargeons le modèle "sentence-transformers/paraphrase-mpnet-base-v2"** pour générer des paires des tweets .

Lors du chargement, le transformateur de phrases effectue automatiquement plusieurs étapes, telles que :

- Conversion des tweets en minuscules.
- Suppression de la ponctuation.
- Lemmatisation ou racinisation des mots.
- Suppression des mots vides (stop words).
- Tokenisation de tous les tweets.
- Encodage de chaque jeton en vecteur d'embedding.

2. Génération des paires de tweets

Dans cette étape, le modèle génère des paires de tweets textuels qui représentent un contenu similaire ou différent en fonction de leur catégorie. Ce processus guide le modèle dans l'apprentissage de représentations significatives. Voici comment cela se déroule :

Après l'encodage des tweets tokenisé la bibliothèque Sentence transformers en utilisant le modèle pré-entraînés "paraphrase-mpnet-base-v2", pour générer des paires de tweets avec leurs classes correspondantes, notées $D = (X, Y)$. Nous avons pris un exemple de la sous-tâche 3 de notre ensemble de données, où X représente l'ensemble des tweets et Y l'ensemble de leurs classes parmi les quatre catégories spécifiées : informations financières, opinion subjective, publicité et annonce. Le modèle génère un ensemble de R triplets positifs $T_{pc} = (x_i, x_j, 1)$ où $x_i = tweet1$ et $x_j = tweet2$ sont choisis aléatoirement parmi ceux de la même classe. De même, il génère également un ensemble de triplets R négatifs $T_{cn} = (x_i, x_j, 0)$, où x_i est un tweet d'une classe Y et x_j est un tweet d'une classe différente. Ensuite, un ensemble d'entraînement T est formé en regroupant les triplets positifs et négatifs des étiquettes de classe : $T = (T_{0p}, T_{0n}), (T_{1p}, T_{1n}), \dots, (T_{|pC|}, T_{|nC|})$, où $|C|$ dénote le nombre d'étiquettes de classe, $|T| = 2R|C|$ est le nombre total de paires dans T , et R est un hyperparamètre.

Ce processus de génération de triplets est crucial pour affiner le modèle pré-entraîné, car il permet au modèle de mieux comprendre les similitudes et les différences entre les tweets appartenant à différentes classes. Les triplets positifs renforcent les représentations sémantiques des tweets similaires, tandis que les triplets négatifs aident à différencier les tweets appartenant à des classes distinctes. Cette méthode permet de créer un ensemble d'entraînement équilibré

et riche en informations, essentiel pour améliorer la performance du modèle lors de la classification des tweets.

3. La phase d'affinement implique l'ajustement progressif des poids du modèle sur des données spécifiques à la tâche, en utilisant les triplets générés. Ce processus vise à optimiser les performances du modèle en ajustant ses paramètres pour mieux représenter les caractéristiques sémantiques des tweets.

2.8.1.3 Entraînement de la tête de classification (Classification Head Training)

Dans cette étape, nous formons une tête de classification en utilisant les embeddings des phrases générés par le modèle ST affiné. Ces embeddings servent d'entrées pour la tête de classification, qui est par défaut une régression logistique dans notre modèle de transformateur de phrases. Voici comment cette étape se déroule :

- Le modèle ST affiné génère des embeddings pour chaque tweet de notre ensemble de données.
- Ces embeddings sont utilisés comme entrée pour la tête de classification, qui est généralement une régression logistique dans notre modèle par défaut.
- La tête de classification est entraînée à associer les embeddings de phrases générés par le modèle ST affiné avec les étiquettes de classe correspondantes extraites de l'ensemble de données.

Ce processus permet à la tête de classification d'apprendre à associer les embeddings, qui représentent le sens du texte, aux étiquettes de classe appropriées.

Dans la phase d'inférence qui suit :

- Pour chaque nouvelle entrée textuelle que nous souhaitons classifier, le modèle ST affiné génère un vecteur d'embedding correspondant. Ce vecteur capture la représentation sémantique du texte.
- Les vecteurs d'embedding générés par le modèle ST affiné pour chaque nouvelle entrée textuelle, encapsulant le sens essentiel du texte, servent d'entrées à la tête de classification. Cette dernière utilise ensuite ces vecteurs pour prédire leur classe correspondante.

Cette approche assure une classification précise des textes en se basant sur leur contenu sémantique, tel que capturé par le modèle ST affiné.

Le tableau 2.1 ci-dessous représente les meilleurs paramètres que nous avons utilisés pour le modèle SetFit :

Hyperparamètres	Valeurs
Batch-size	32
Num_epochs	9
Num_ iterations	10
Metric	Accuracy, F1-score

TABLE 2.1 – Les détails des techniques.

2.8.2 DistilBERT

DistilBERT est une version simplifiée , allégée et plus rapide de BERT, le modèle original de traitement du langage naturel. DistilBERT conserve l’architecture de base de BERT tout en réduisant sa taille de 40% [Vaswani et al.,2017]et en augmentant sa vitesse de traitement de 60% [Sanh, Debut, Chaumond, et Wolf,2019], tout en conservant 97% [Sanh et al.,2019] de ses capacités de compréhension linguistique. Cette simplification est réalisée en retirant les embeddings token-type et le pooler, et en réduisant de moitié le nombre de couches. L’accent est également mis sur une utilisation efficace du réseau pour assurer une convergence rapide du modèle. DistilBERT se présente ainsi comme une alternative prometteuse à BERT, offrant une meilleure efficacité en termes de vitesse de traitement et de consommation de ressources sans compromettre les performances linguistiques [Sanh et al.,2019].

Architecture du DistilBERT

DistilBERT est un modèle de transformateurs plus petit et plus rapide que BERT. Il a été introduits en 2019 par Victor Sanh, Lysandre Debut, Julien Chaumond et Thomas Wolf dans leur article [Face,2023a]. Il a été pré-entraîné de manière auto-supervisée sur le même corpus que BERT, en utilisant ce dernier comme modèle enseignant. Cela signifie que le pré-entraînement s’est fait uniquement sur des textes bruts, sans intervention humaine pour étiqueter les données. Ce processus automatique permet de générer des entrées et des étiquettes à partir de ces textes en s’appuyant sur le modèle de base BERT, ce qui lui permet d’exploiter un grand nombre de données accessibles au public [Face,2023a].

DistilBERT a montré des performances compétitives dans des tâches basées sur des textes telles que la modélisation linguistique et l’analyse des sentiments. Par conséquent, ce modèle est particulièrement bien adapté pour la tâche de classification de texte, qui nécessite une analyse approfondie de l’ensemble du texte pour attribuer des catégories ou des labels spécifiques en fonction du contenu et du contexte. La figure 2.7 présente l’architecture de DistilBERT.Cette figure montre un exemple de la sous-tâche 3 de notre ensemble de données, les deux autres sous-tâches étant similaires.

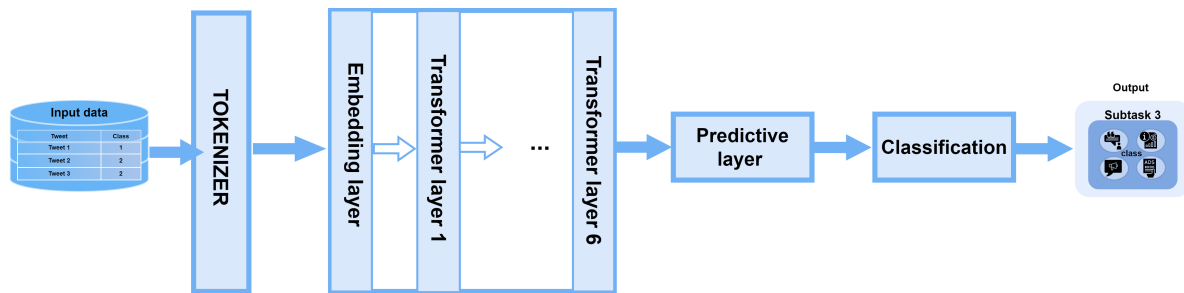


FIGURE 2.7 – Architecture du DistilBERT.

D’abord, nous avons nettoyé les tweets. Ensuite, le modèle DistilBERT a effectué la tokenisation et l’encodage de ces tweets en utilisant son tokenizer spécifique, basé sur une méthode efficace de tokenisation par sous-mots.

Ce processus génère des vecteurs contenant des jetons, chacun ayant un identifiant unique. Le vecteur contient également des jetons spéciaux intégrés dans le texte saisi. Le jeton [CLS], avec une valeur de 101, est ajouté au début du texte pour marquer le début de la phrase, tandis que le jeton [SEP], représenté par la valeur 102, est ajouté après chaque phrase pour indiquer la fin de celle-ci. De plus, un masque d’attention, constitué d’un vecteur binaire de la même longueur que la séquence de jetons, est utilisé pour indiquer quels jetons sont pertinents pendant l’entraînement. L’affinement du modèle DistilBERT pour la classification des tweets implique un ajustement progressif des poids du modèle en utilisant des données spécifiques à la tâche.

L’objectif de ce processus est d’améliorer les performances du modèle en ajustant ses paramètres pour mieux représenter les caractéristiques sémantiques des tweets. Le modèle affiné génère ensuite des représentations latentes (The latent representation) de 768 dimensions pour chaque tweet à travers ses six couches de transformation. Ces couches utilisent des mécanismes d’attention multi-têtes et des réseaux neuronaux feedforward pour encoder les informations des jetons en embeddings denses, riches en informations sémantiques et contextuelles.

Enfin, une couche de classification est située à la fin du modèle DistilBERT. Cette couche utilise les représentations latentes des tweets, générées par le modèle DistilBERT affiné, comme entrées et s’entraîne avec ces représentations. Lorsqu’un nouveau tweet est introduit, le modèle affiné génère les représentations latentes pour ce tweet, qui sont ensuite utilisées par la couche de classification pour prédire la classe correspondante. Une fonction d’activation Softmax est appliquée sur les sorties de cette couche pour produire des probabilités pour chaque classe de sortie, permettant ainsi une prédiction précise des classifications, avec les scores représentant la confiance du modèle pour chaque classe.

Nous détaillons dans ce qui suit, chacune de ces étapes :

2.8.2.1 Chargement du modèle

Pour charger le modèle DistilBERT pour la classification des tweets, nous l'avons utilisé les deux instructions suivantes :

- **AutoTokenizer** : La fonction AutoTokenizer de la bibliothèque Transformers est responsable du chargement du tokenizer. Il est important de choisir soigneusement le tokenizer, car différents transformateurs peuvent avoir différentes méthodes de tokenisation (algorithmes) ou interpréter les jetons différemment. Assurer la compatibilité et comprendre le processus de tokenisation est crucial lorsque l'on travaille avec des transformateurs.
- **DistilBertForSequenceClassification** : une classe fournie par la bibliothèque Hugging Face Transformers ⁶. Il s'agit d'un modèle pré-entraîné basé sur l'architecture DistilBERT, adapté aux tâches où l'entrée est une séquence de texte (par exemple, une phrase, un paragraphe) et la sortie est une étiquette de classification (par exemple, sentiment positif/négatif, catégorie de sujet).

2.8.2.2 Tokenisation des tweets à l'aide du tokenizer "Distilbert"

La première opération nécessaire avant de transmettre des données au transformateur est la tokenisation. Les transformateurs ne peuvent traiter que des nombres, la tokenisation est donc le processus de conversion des données textuelles en jetons compréhensibles par les modèles. Dans ce cas, nous avons utilisé le tokenizer "DistilBERT". Il existe trois types courants de tokenisation utilisés par les transformateurs : basée sur les mots, basée sur les caractères et basée sur les sous-mots [Face,2023b].

Le tokenizer DistilBERT utilise la méthode de tokenisation basée sur les sous-mots, qui offre un avantage de vitesse jusqu'à 4 fois plus rapide par rapport au tokenizer classique qui utilise une méthode de tokenisation basée sur les mots.

Ce processus génère deux vecteurs de résultats :

Le masque d'attention(The attention mask) Un élément crucial des données d'entrée générées lors du processus de tokenisation. Il s'agit essentiellement d'un vecteur binaire de même longueur que la séquence d'entrée. Son rôle consiste à indiquer quels tokens le modèle doit prendre en compte lors de l'entraînement et de l'inférence. Les valeurs du masque sont définies à 1 pour les tokens valides et à 0 pour les tokens de remplissage ajoutés à la séquence afin d'uniformiser sa longueur.

Les input_ids(The input_ids) Le vecteur de résultat contient des jetons, où

6. <https://huggingface.co/>

chaque mot, caractère ou sous-mot est représenté par son propre jeton, de manière similaire à un dictionnaire.

Comme mentionné précédemment, les modèles ne peuvent traiter que des nombres. Le vecteur contient également des jetons spéciaux tels que [CLS] et [SEP] intégrés dans le texte saisi. Le jeton [CLS] est employé pour marquer le début de la phrase, tandis que [SEP] est utilisé pour indiquer la fin d'une phrase ou pour séparer différentes parties de l'entrée textuelle lors du processus de tokenisation et d'encodage par DistilBERT.

Ces jetons spéciaux sont essentiels pour permettre au modèle de transformer et de comprendre la structure du texte en entrée et d'effectuer des prédictions précises.

2.8.2.3 Mise au point du modèle (Fine-tuning of the model)

Le modèle DistilBERT pré-entraîné a été affiné pour la classification des tweets afin d'optimiser progressivement ses poids en utilisant des données spécifiques à cette tâche. L'objectif principal de ce processus est d'améliorer les performances du modèle en ajustant ses paramètres pour mieux capturer les caractéristiques sémantiques des tweets.

2.8.2.4 Le générateur des représentation latente (The latent representation generator)

Extrait des vecteurs de 768 dimensions pour chaque séquence de notre jeu de données après avoir ajusté le modèle. Ces représentations capturent les aspects essentiels du texte d'origine et sont exprimées sous forme de vecteurs numériques réels.

2.8.2.5 Formation des classificateurs (Training of classifiers)

Dans la dernière étape de notre traitement, une couche de classification est située à la fin du modèle DistilBERT.

Cette couche utilise comme entrées les représentations latentes des tweets, générées par les couches de transformation du modèle DistilBERT affiné, et s'entraîne avec ces représentations.

Lorsqu'un nouveau texte est introduit, le modèle fine-tuné génère ses représentations latentes. Ces représentations sont ensuite utilisées par la couche de classification pour prédire la classe correspondante du texte.

La fonction d'activation Softmax est appliquée sur les sorties de cette couche pour produire des probabilités pour chaque classe de sortie, facilitant ainsi une prédiction précise des classifications. Les scores obtenus représentent la confiance du modèle pour chaque classe.

Le tableau 2.2 ci-dessous représentent les meilleures paramètre qui nous avons utilisé pour le modèle DistilBERT :

Hyperparamètres	Valeurs
max_seq_length	512
word_embedding_dimension	768
Learning_rate	7e-3
num_hidden_layers	6
max_position_embeddings	512
num_train_epochs	10

TABLE 2.2 – Les détails des techniques.

2.9 Conclusion

Dans ce chapitre, nous commençons par présenter en détail l'ensemble de données utilisé, suivi de la description des différentes étapes de préparation des données et des techniques de prétraitement mises en œuvre pour nettoyer les données. Ensuite, nous détaillons les différents modèles d'apprentissage par transfert proposés, ainsi que leurs architectures respectives.

Dans le chapitre suivant, nous décrirons le dispositif expérimental et les résultats obtenus pour chaque modèle. Nous détaillerons également les outils et bibliothèques utilisés, et comparerons les performances de nos modèles avec celles des approches de l'état de l'art.

Chapitre 3

Expérimentation et résultats

3.1 Introduction

Dans ce chapitre, les modèles développés pour classifier les tweets des influenceurs sont analysés en détail dans le contexte du profilage des auteurs . Leurs performances sont examinées et comparées aux résultats de l'état de l'art.

Pour cela, diverses mesures d'évaluation sont utilisées, qui seront également présentées. De plus, les expériences menées ainsi que les outils et les bibliothèques qui ont été utiles au cours du travail sont décrits.

3.2 Environnement du logiciel

Pour nos expériences, nous avons utilisé Python 3.10 comme langage de programmation dans l'environnement Kaggle, dont nous parlerons dans cette section, ainsi que les diverses bibliothèques Python qui ont été essentielles dans notre travail.

3.2.1 Kaggle

Notre environnement de travail était le Kaggle Notebook sur le site web de Kaggle. Kaggle Notebook est un environnement informatique basé sur le web pour l'analyse de données, l'apprentissage automatique et les tâches d'apprentissage profond qui permet aux utilisateurs de travailler ensemble pour collaborer à un projet commun. Il inclut des bibliothèques de science des données bien connues en tant que dépendances pré-installées. Les cellules de code peuvent être utilisées par les utilisateurs pour écrire du code interactif. Il s'agit globalement d'un environnement complet et convivial pour l'analyse des données, le développement de modèles et le partage des connaissances.

3.2.2 Bibliothèques Python

Dans cette section, nous présentons les bibliothèques Python qui ont joué un rôle crucial dans notre travail. Python, étant un langage de programmation puissant, propose une grande variété de bibliothèques qui ont facilité la conception et la mise en œuvre de nos modèles.

3.2.2.1 NumPy

NumPy¹ est une bibliothèque Python open source et gratuite, conçue pour le calcul scientifique et l'analyse de données. Elle offre une prise en charge efficace des grands tableaux multidimensionnels, accompagnée d'une gamme étendue de fonctions mathématiques avancées. NumPy constitue la base de nombreuses autres bibliothèques importantes comme Pandas, Scikit-learn et SciPy, facilitant ainsi les opérations mathématiques complexes et optimisées sur des ensembles de données volumineux.

3.2.2.2 Pandas

Pandas² est la bibliothèque open-source la plus largement utilisée pour la manipulation et l'analyse de données, construite sur le langage de programmation Python. Elle offre des structures de données et des opérations pour une analyse et une manipulation de données solides, flexibles et conviviales. Pandas enrichit Python en lui permettant de travailler avec des données similaires à des feuilles de calcul, facilitant le chargement rapide, l'alignement, la manipulation et la fusion, ainsi que d'autres opérations cruciales.

3.2.2.3 Scikit-learn

Scikit-learn³ est une bibliothèque d'apprentissage automatique très réputée et largement utilisée en Python. Elle offre une multitude d'algorithmes pour la classification, la régression et le clustering, ainsi que des outils essentiels pour la visualisation, le prétraitement des données, et l'évaluation des modèles. Basée sur NumPy, SciPy et matplotlib, Scikit-learn se distingue par sa facilité d'utilisation grâce à des API bien conçues, une documentation complète et une communauté active qui fournit de nombreux tutoriels et exemples pratiques.

3.2.2.4 Transformers

La bibliothèque Transformers⁴, maintenue par Hugging Face et la communauté, simplifie le téléchargement et l'entraînement de modèles d'apprentissage automatique pré-entraînés avancés. Compatible avec PyTorch, TensorFlow et JAX, elle prend en charge une large

1. <https://www.nvidia.com/en-us/glossary/numpy/>

2. <https://www.nvidia.com/en-us/glossary/pandas-python/>

3. <https://www.nvidia.com/en-us/glossary/scikit-learn/>

4. <https://klu.ai/glossary/transformers-library>

gamme de tâches dans le traitement du langage naturel, la vision par ordinateur, l'audio et les tâches multimodales. Elle offre une API conviviale pour l'inférence via la méthode pipeline, ainsi que des milliers de modèles pré-entraînés accessibles pour diverses modalités, facilitant ainsi le développement d'applications complexes en apprentissage automatique.

3.3 Mesures d'évaluation

Avant de définir chaque métrique, nous devons d'abord établir une compréhension claire des principes fondamentaux utilisés dans ces métriques, en utilisant les quatre classes de la sous-tâche 3 de notre ensemble de données : informations financières, opinion subjective, publicité et annonce.

- **TP (True Positives)** : Les vrais positifs font référence aux instances qui sont correctement prédites comme appartenant à une classe donnée. Par exemple, si un texte appartient à la classe informations financières et que le modèle prédit informations financières, c'est un vrai positif pour cette classe .
- **TN (True Negatives)** : Les vrais négatifs font référence aux instances qui sont correctement prédites comme n'appartenant pas à une classe donnée. Par exemple, si un texte n'appartient pas à la classe opinion subjective et que le modèle prédit une classe autre que opinion subjective, c'est un vrai négatif pour la classe opinion subjective.
- **FP (False Positives)** : Les faux positifs représentent les cas où les instances d'autres classes ont été incorrectement classées comme appartenant à la classe donnée. Par exemple, si un texte appartient à la classe publicité mais que le modèle prédit opinion subjective, c'est un faux positif pour la classe opinion subjective.
- **FN (False Negatives)** : Les faux négatifs font référence aux cas qui, bien qu'ils appartiennent à la classe donnée, ont été faussement prédits comme appartenant à une autre classe. Par exemple, si un texte appartient à la classe publicité mais que le modèle prédit annonce, c'est un faux négatif pour la classe publicité.

3.3.1 Accuracy

L'accuracy mesure le pourcentage de profils d'influenceurs correctement prédits par le modèle. Nous avons utilisé cette mesure pour évaluer les performances de nos modèles. C'est défini comme suit :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Dans un contexte multi-classes, l'Accuracy est la proportion de toutes les prédictions correctes sur l'ensemble des classes.

3.3.2 F1-score

F1-score est la mesure harmonique entre la précision (Precision) et le rappel (Recall) [M et M.N,2015].

Pour une classe donnée, la précision est le nombre de vrais positifs divisé par le nombre total de prédictions positives, et le rappel est le nombre de vrais positifs divisé par le nombre total d'instances réellement positives. F1-score est calculé (par exemple, on remplace la classe annonce par A) comme suit :

$$\begin{aligned} \text{Precision}_A &= \frac{TP_A}{TP_A + FP_A} \\ \text{Recall}_A &= \frac{TP_A}{TP_A + FN_A} \\ \text{F1}_A &= 2 \cdot \frac{\text{Précision}_A \cdot \text{Rappel}_A}{\text{Précision}_A + \text{Rappel}_A} \end{aligned}$$

3.4 Configuration expérimentale

Dans cette section, nous présentons les expériences et les résultats des modèles proposés.

L'objectif est d'évaluer les performances de chaque modèle et d'explorer l'impact de l'utilisation de tweets non traités par rapport aux tweets prétraités.

À la fin des expériences, nous fournissons une comparaison entre les résultats obtenus par nos modèles et ceux obtenus par d'autres modèles dans l'état de l'art.

3.4.1 Tweets purs vs tweets prétraités

Dans cette expérience, nous évaluons les deux modèles SetFit et DistilBERT. Nous commençons par tester avec des tweets bruts, puis nous passons aux tweets prétraités.

3.4.1.1 Utilisation des tweets bruts

Dans cette expérience, nous évaluons les modèles en utilisant les tweets dans leur état bruts, tels qu'ils sont publiés dans notre ensemble de données, sans effectuer de prétraitement préalable. Nous présentons ensuite les résultats obtenus pour chaque modèle en utilisant les différentes métriques mentionnées précédemment.

■ Résultats du modèle DistilBERT :

Les résultats obtenus par le modèle DistilBERT sont présentés dans le tableau représenté 3.1 ci-dessous.

Pour ce modèle, nous avons utilisé "adamw_torch" comme optimiseur, et un nombre d'époques égal à "10".

Mesures	Sous-tâches		
	Sous-tâche1	Sous-tâche2	Sous-tâche3
F1-score	57.77%	58.03%	74.39%
Accuracy	62.5%	57.81%	75%

TABLE 3.1 – Les performances du DistilBERT sur les tweets bruts en utilisant diverses mesures d'évaluation pour les trois sous-tâches.

SOUS-TÂCHE 1 : Le modèle a réalisé un F1-score de 57,77% et une perte égale à 1,21.

Les figures 3.1 et 3.2 présentent le F1-score et les valeurs de perte au fil des époques pour les données d'entraînement et de validation :

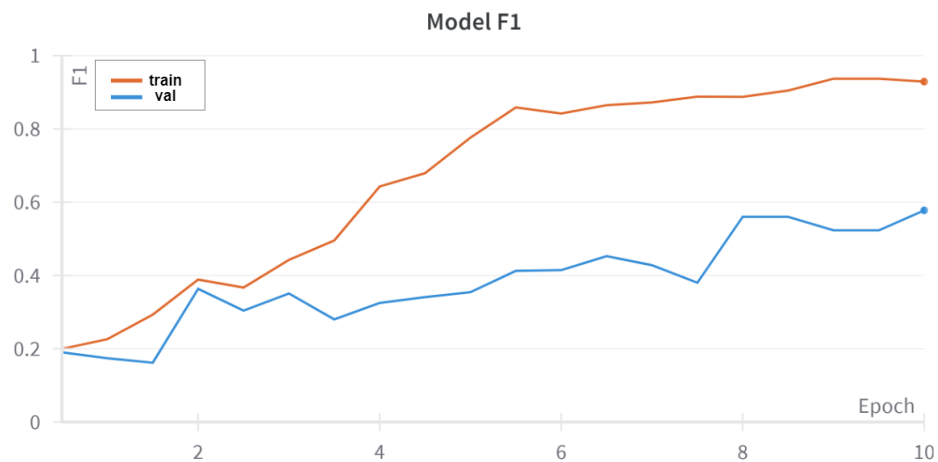


FIGURE 3.1 – Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets bruts.



FIGURE 3.2 – Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets bruts.

SOUS-TÂCHE 2 : Le modèle a atteint un F1-score de 58,03% et une perte égale à 1,417.

Les figures 3.3 et 3.4 illustrent le F1-score et les valeurs de perte à travers les époques pour l'entraînement et la validation des données :

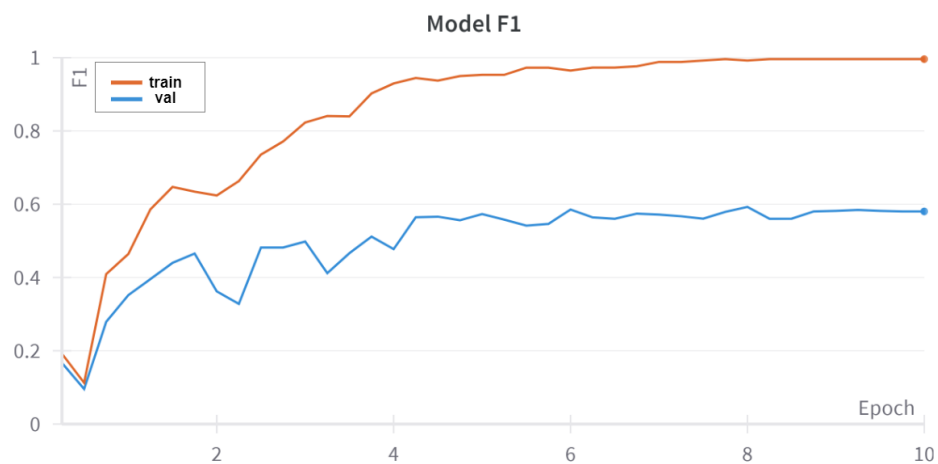


FIGURE 3.3 – Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets bruts.



FIGURE 3.4 – Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets bruts.

SOUS-TÂCHE 3 : Le modèle a obtenu un F1-score de 74,39 % et une perte de 0,795.

Les figures 3.5 et 3.6 montrent respectivement l'évolution du F1-score et des valeurs de perte au fil des époques pour l'entraînement et la validation des données :

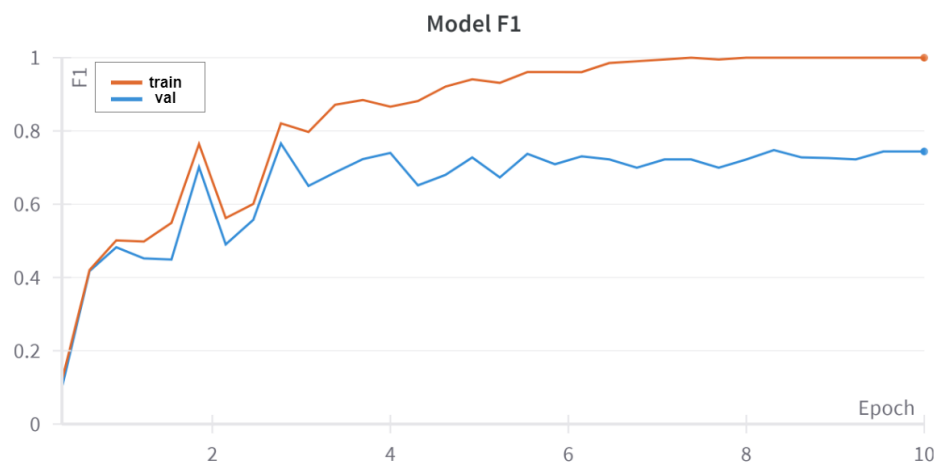


FIGURE 3.5 – Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets bruts.



FIGURE 3.6 – Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets bruts.

■ RÉSULTATS DU MODÈLE SETFIT :

Les résultats des différentes mesures d'évaluation (F1-score et accuracy) pour les trois sous-tâches avec le modèle SetFit sont présentés dans le tableau 3.2 ci-dessous. Pour ce modèle, nous avons utilisé "CosineSimilarityLoss" comme fonction de perte.

Sous-tâches \ Mesures	Sous-tâche1	Sous-tâche2	Sous-tâche3
F1-score	44,32 %	62,43%	82,23%
Accuracy	46,88 %	62,5%	82,69%

TABLE 3.2 – Les performances du SetFit sur les tweets bruts en utilisant diverses mesures d'évaluation pour les trois sous-tâches.

SOUS-TÂCHE 1 :

Pour cette sous-tâche, le modèle n'a été exécuté que pendant 4 époques en raison des exigences élevées en mémoire de l'environnement d'exécution. Le modèle a obtenu un F1-score de 44,32%. La figure 3.7 illustre l'évolution du F1-score au fil des époques pour les données de validation.

SOUS-TÂCHE 2 :

Pour cette sous-tâche, nous avons utilisé un nombre d'époques égal à 8. Le modèle a atteint un F1-score de 62,43%. La figure 3.7 montre l'évolution du F1-score au fil des époques pour les données de validation.

SOUS-TÂCHE 3 :

Pour cette sous-tâche, nous avons utilisé un nombre d'époques égal à 9. Le modèle a réalisé un F1-score de 82,23%. La figure 3.7 présente l'évolution du F1-score au fil des époques pour les données de validation.

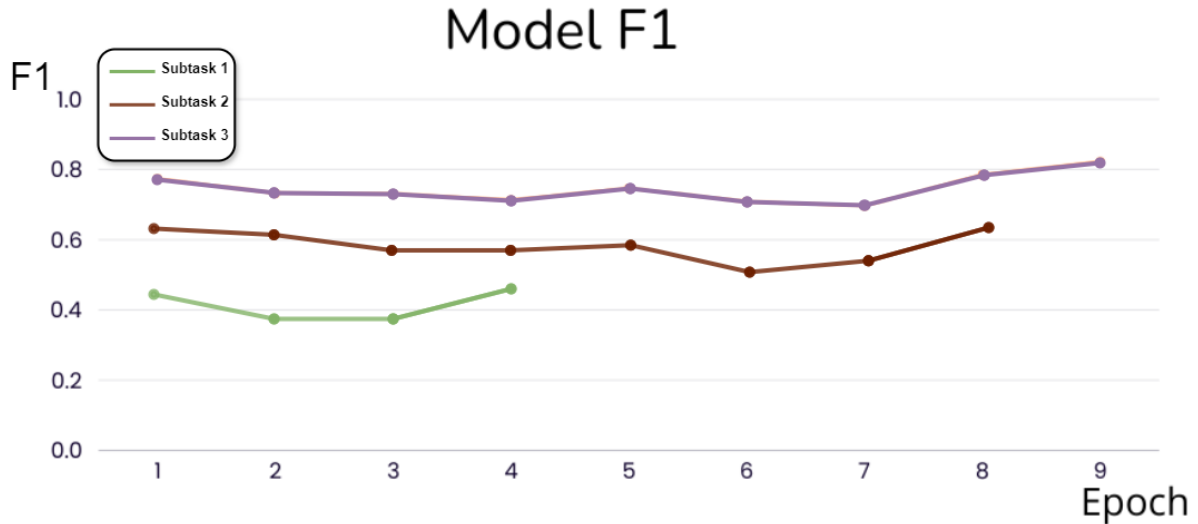


FIGURE 3.7 – Le graphe du F1-score des trois sous-tâches utilisant SetFit sur des tweets burts.

3.4.1.2 Utilisation des tweets prétraités

Dans cette expérience, nous appliquons les étapes de prétraitement détaillées dans le chapitre 2 à nos données. Ensuite, nous évaluons les performances des deux modèles et présentons les résultats obtenus en utilisant les différentes métriques mentionnées précédemment.

■ **RÉSULTATS DU MODÈLE DISTILBERT** : Les performances du modèle DistilBERT pour les trois sous-tâches sont présentées dans le tableau 3.3 ci-dessous. Pour ce modèle, nous avons utilisé "adamw_torch" comme optimiseur, et un nombre d'époques égal à "10".

Mesures	Sous-tâches		
	Sous-tâche1	Sous-tâche2	Sous-tâche3
F1-score	53%	54,56%	80,36%
Accuracy	53,12%	54,63%	80,77%

TABLE 3.3 – Les performances du DistilBERT sur les tweets prétraité en utilisant diverses mesures d'évaluation pour les trois sous-tâches.

SOUS-TÂCHE 1 : Le modèle a obtenu un F1-score de 53% et une perte de 1,15. Les figures 3.8 et 3.9 montrent l'évolution du F1-score et des valeurs de perte au fil des époques pour les ensembles d'entraînement et de validation, respectivement :

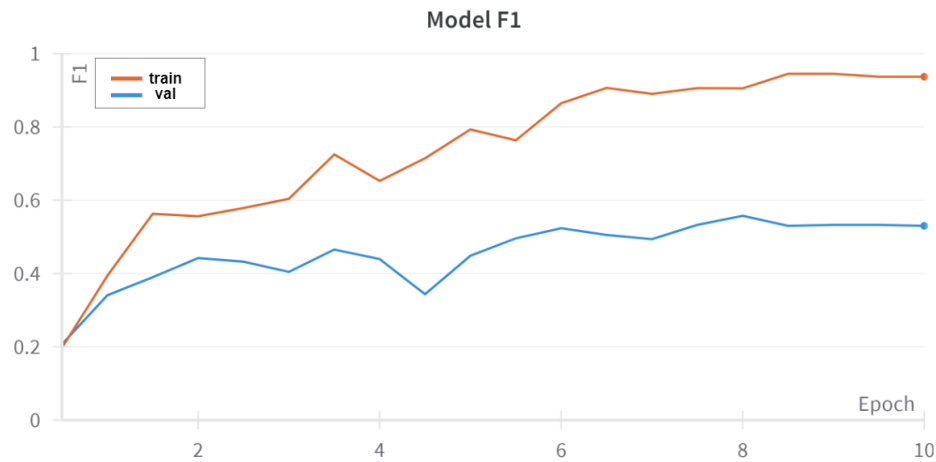


FIGURE 3.8 – Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets prétraités.



FIGURE 3.9 – Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets prétraités.

SOUS-TÂCHE 2 : Le modèle a atteint un F1-score de 54,56% et une perte égale à 1,43.

les figures 3.10 et 3.11 illustrent F1-score et les valeurs de perte à travers les époques pour l'entraînement et la validation données :

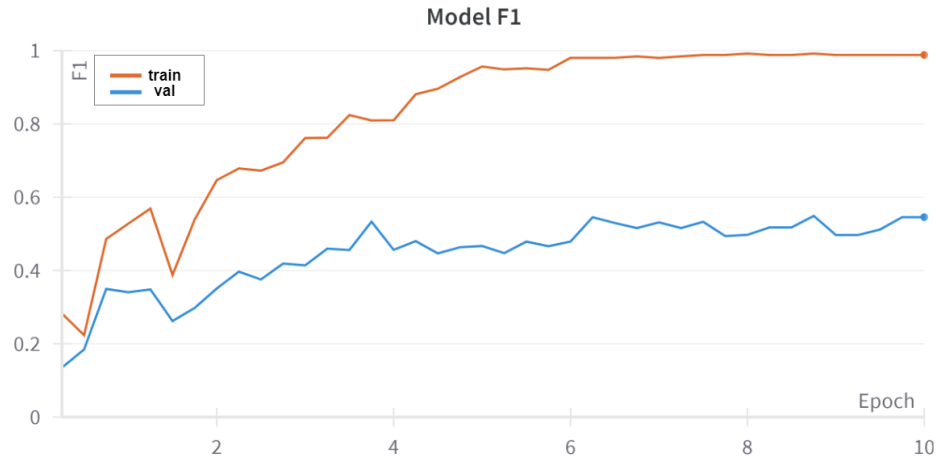


FIGURE 3.10 – Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets prétraités.



FIGURE 3.11 – Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets prétraités.

SOUS-TÂCHE 3 : Le modèle a réalisé un F1-score de 80,36% et une perte de 0,7036.

Les figures 3.12 et 3.13 présentent respectivement l'évolution du F1-score et des valeurs de perte au cours des époques pour les ensembles de données d'entraînement et de validation :

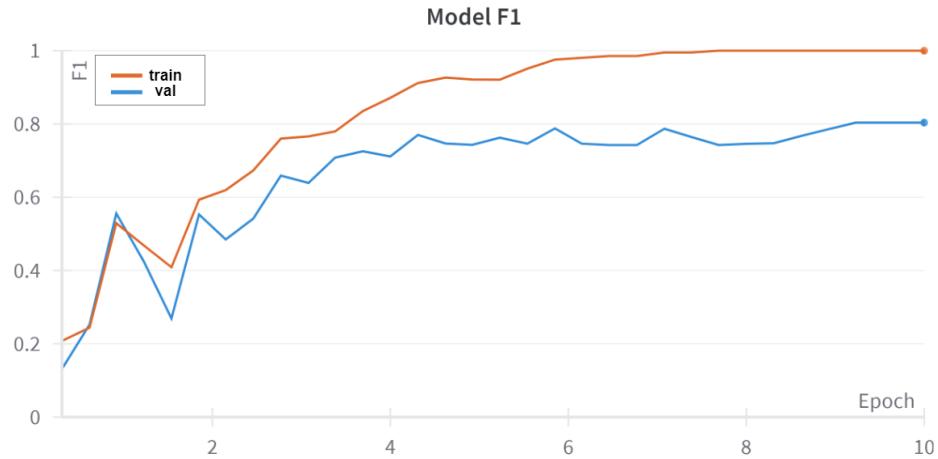


FIGURE 3.12 – Le graphe du F1-score avec 10 époques utilisant DistilBERT sur des tweets prétraités.

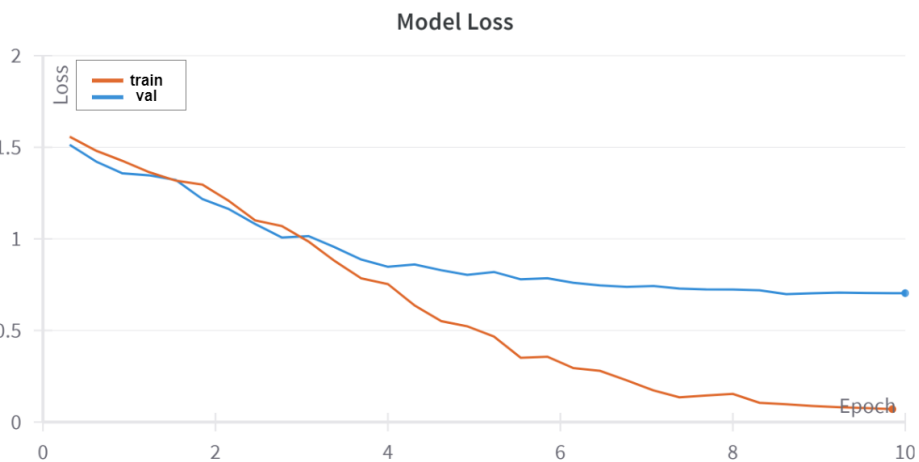


FIGURE 3.13 – Le graphe des pertes avec 10 époques utilisant DistilBERT sur des tweets prétraités.

■ RÉSULTATS DU MODÈLE SETFIT :

Les résultats des différentes mesures d'évaluation obtenus par le modèle SetFit pour les trois sous-tâches sont présentés dans le tableau 3.4 représenté ci-dessous .

Pour ce modèle, nous avons utilisé "CosineSimilarityLoss" comme fonction de perte.

Mesures	Sous-tâches		
	Sous-tâche1	Sous-tâche2	Sous-tâche3
F1-score	56%	57,5%	75,04%
Accuracy	56,25%	57,81%	75%

TABLE 3.4 – Les performances du SetFit sur les tweets prétraité en utilisant diverses mesures d'évaluation pour les trois sous-tâches.

SOUS-TÂCHE 1 :

Pour cette sous-tâche, le modèle n'a été exécuté que pendant 4 époques en raison des exigences élevées en mémoire de l'environnement d'exécution. Le modèle a obtenu un F1-score de 56%. La figure 3.14 présente l'évolution du F1-score au fil des époques pour les ensembles de données de validation

SOUS-TÂCHE 2 :

Pour cette sous-tâche, nous avons utilisé un nombre d'époques égal à 8. Le modèle a atteint un F1-score de 57,5%. La figure 3.14 illustre l'évolution du F1-score au cours des époques pour les ensembles de données de validation.

SOUS-TÂCHE 3 :

Pour cette sous-tâche, nous avons utilisé un nombre d'époques égal à 9. Le modèle a réalisé un F1-score de 75,04%. La figure 3.14 montre l'évolution du F1-score au cours des époques pour les ensembles de données de validation.

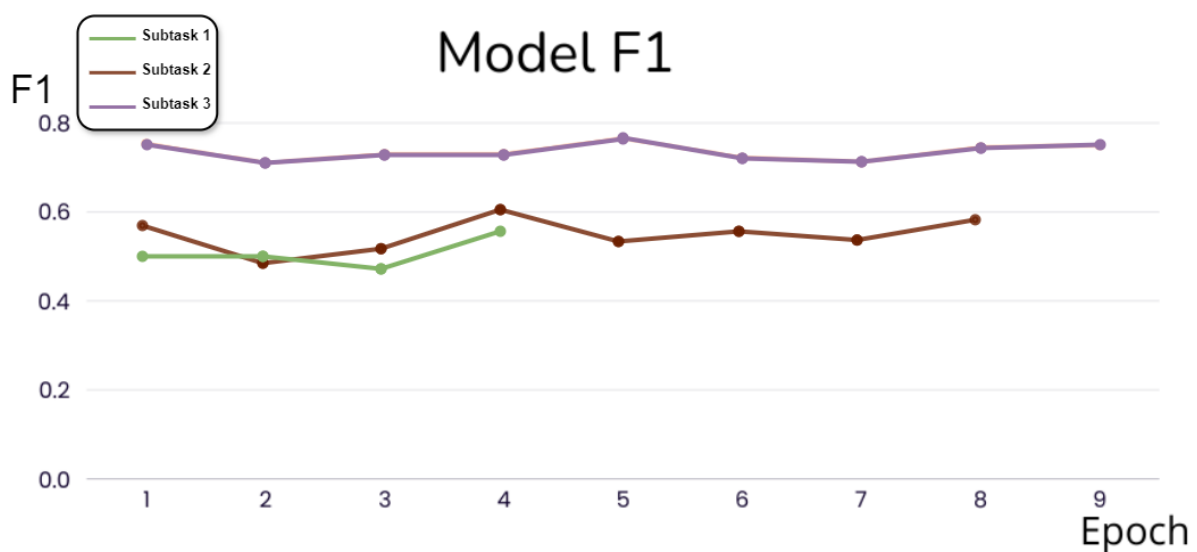


FIGURE 3.14 – Le graphe du F1-score des trois sous-tâches utilisant SetFit sur des tweets prétraités.

3.4.2 Analyse comparative des performances des deux modèles

Le tableau 3.5 ci-dessous présente un résumer de tous les résultats qui nous avons obtenu pour les deux modèles avec et sans prétraitement.

Modèles	Tweets burts		Tweets prétraités	
	DistilBERT	SetFit	DistilBERT	SetFit
Sous-tâches				
Sous-tâche1 - Profilage des influenceurs à faibles ressources	57,77%	44,32%	53%	56%
Sous-tâche2 - Identification des intérêts des influenceurs disposant de peu de ressources	58,03%	62,43%	54,56%	57,5%
Sous-tâche3 - Identification des intentions des influenceurs à faibles ressources	74,39%	82,23%	80,36%	75,04%

TABLE 3.5 – Les résultats des deux modèles sont comparés en termes de F1-score, avec et sans prétraitement.

SOUS-TÂCHE 1 :

— *Avant Prétraitement*

— **DistilBERT : 57,77%**

— **SetFit : 44,32%**

Pour cette sous-tâche, avant le prétraitement, DistilBERT surpasse SetFit avec un score de 57,77% contre 44,32%. Cela suggère que DistilBERT est plus efficace que SetFit pour identifier les influenceurs des cryptomonnaies en fonction de leur degré d'influence, lorsque aucun prétraitement n'est appliqué.

— *Après Prétraitement*

— **DistilBERT : 53%**

— **SetFit : 56%**

Après le prétraitement, les performances de DistilBERT diminuent légèrement à 53%, tandis que celles de SetFit s'améliorent significativement pour atteindre 56%. Ce changement montre que le prétraitement bénéficie davantage à SetFit qu'à DistilBERT pour cette sous-tâche, inversant ainsi les résultats obtenus avant le prétraitement.

SOUS-TÂCHE 2 :

— *Avant Prétraitement*

— **DistilBERT : 58,03%**

— **SetFit : 62,43%**

Pour cette sous-tâche, SetFit surpasse DistilBERT avec un score de 62,43% contre 58,03% avant le prétraitement. Cela montre que SetFit est plus efficace pour identifier les intérêts des influenceurs lorsque les tweets ne sont pas prétraités.

— *Après Prétraitement*

— **DistilBERT : 54,56%**

— **SetFit : 57,5%**

Après prétraitement, les performances de DistilBERT et SetFit diminuent toutes les deux, mais SetFit conserve son avantage avec 57,5% contre 54,56% pour DistilBERT. Cela suggère que le prétraitement réduit légèrement l'efficacité des deux modèles pour cette sous-tâche, bien que SetFit reste le meilleur des deux.

SOUS-TÂCHE 3 :

— *Avant Prétraitement*

— **DistilBERT : 74,39%**

— **SetFit : 82,23%**

Pour l'identification des intentions des influenceurs, SetFit surpasse largement DistilBERT avant le prétraitement, avec un score de 82,23% contre 74,39%. Cette différence significative montre la robustesse de SetFit pour cette sous-tâche spécifique.

— *Après Prétraitement*

— **DistilBERT : 80,36%**

— **SetFit : 75,04%**

Après le prétraitement, les performances de DistilBERT s'améliorent de manière significative pour atteindre 80,36%, tandis que celles de SetFit diminuent légèrement à 75,04%. Ici, le prétraitement semble bénéficier plus à DistilBERT, rendant ses performances presque comparables à celles de SetFit avant le prétraitement.

3.4.3 Comparaison des résultats avec l'état de l'art

Dans cette section, nous effectuons une comparaison entre les résultats obtenus par nos deux modèles et ceux de l'état de l'art. Les résultats sont résumés dans le tableau 3.6.

MODELES \ SOUS-TÂCHES	SOUS-TÂCHE1	SOUS-TÂCHE2	SOUS-TÂCHE3
BERT [Espinosa et Sidorov,2023]	86,77%	83,44%	85,01%
SVM [Ferri-Molla et Santamaria-Jorda,2023]	57%	43%	61%
Perceptron [Ferri-Molla et Santamaria-Jorda,2023]	52%	39%	58%
Régression logistique [Ferri-Molla et Santamaria-Jorda,2023]	57%	43%	43%
SetFit	56%	63,43%	82,23%

TABLE 3.6 – Analyse comparative des résultats par rapport à l’état de l’art en utilisant F1-score.

La comparaison des performances des différents modèles de pointe présentés dans le tableau révèle des insights significatifs. Le modèle BERT se distingue particulièrement en affichant des F1-scores élevés sur toutes les sous-tâches, mettant en avant son efficacité dans la résolution des défis de profilage d’auteurs. En revanche, les modèles SVM, Perceptron et Régression logistique présentent des performances inférieures, suggérant qu’ils pourraient ne pas être aussi adaptés pour ces tâches spécifiques qui exigent une compréhension fine du langage naturel. Concernant notre modèle SetFit développé, il montre des performances compétitives par rapport à l’état de l’art, avec des F1-scores respectables sur toutes les sous-tâches évaluées. Cependant, pour optimiser ses performances, des ajustements supplémentaires dans la méthodologie ou les hyperparamètres du modèle pourraient être nécessaires.

3.5 Interface d’application

Dans cette section, nous présentons l’interface de notre application de profilage des auteurs, la première figure 3.15 illustre l’interface de l’application au lancement :

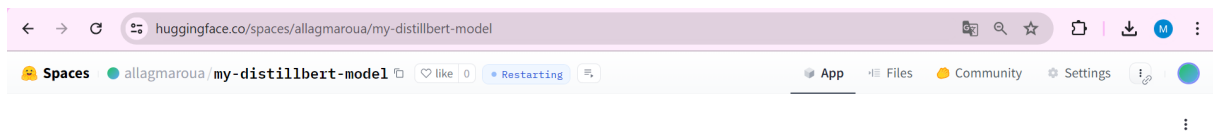


FIGURE 3.15 – L’interface au lancement de l’application.

À partir de ce point, l’utilisateur choisit l’une des trois options illustrées dans la figure 3.16 pour classifier l’influenceur selon ses besoins.

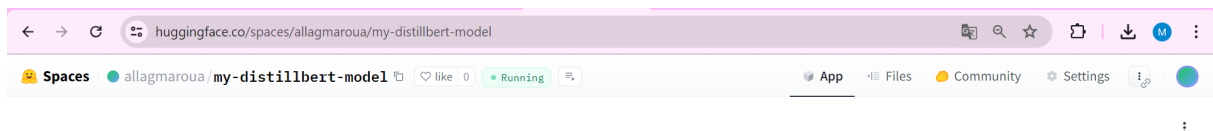


FIGURE 3.16 – L’interface pour choisir le type de profilage du l’influenceur.

Ensuite saisit un tweet dans le champ de texte. Pour prédire l’influenceur, Enfin, l’utilisateur doit cliquer sur le bouton « Classify » et le résultat s’affichera en dessous. La figure 3.17 suivante présente un exemple de prédiction de l’influenceur d’un tweet

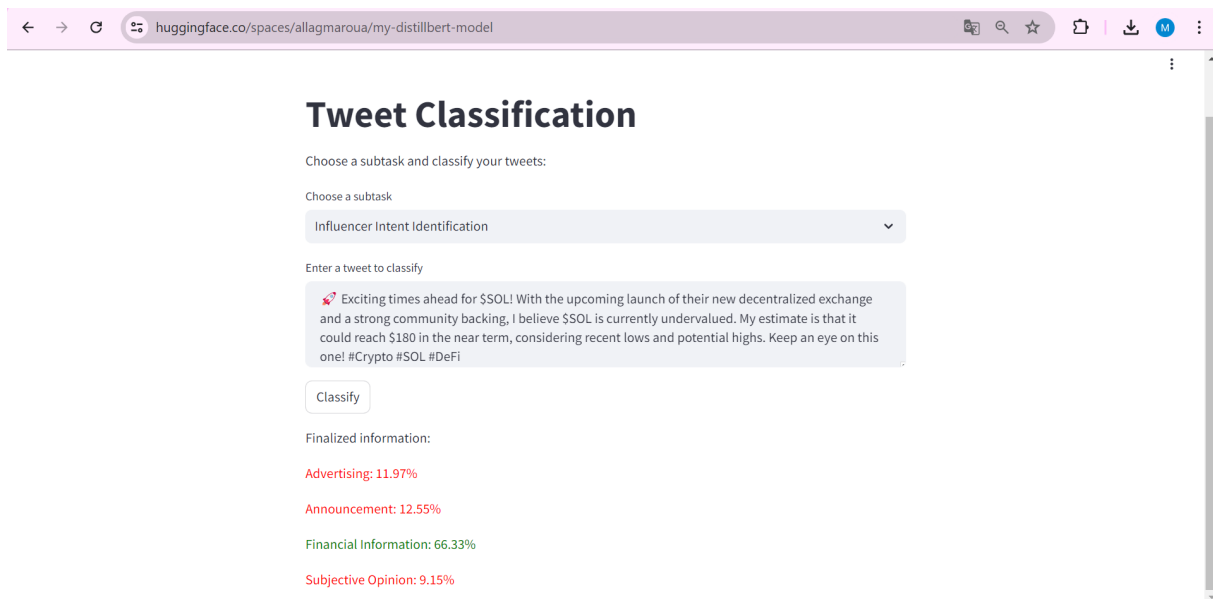


FIGURE 3.17 – L’interface avec un exemple de profilage de l’influenceur d’un tweet.

3.6 Conclusion

En conclusion, dans ce chapitre, nous avons présenté les différentes bibliothèques utilisées dans les deux modèles, ainsi que les deux métriques d’évaluation employées pour mesurer leurs performances. Nous avons ensuite détaillé les expériences menées avec les différents scénarios, fournissant les résultats obtenus pour chacun des deux modèles. De plus, nous avons comparé nos modèles proposés avec les résultats de l’état de l’art, mettant en évidence la complexité de la tâche de profilage des influenceurs de cryptomonnaie et les ressources limitées de notre ensemble de données à travers les résultats obtenus. Enfin, nous avons présenté notre interface utilisateur, illustrant son fonctionnement à travers un exemple de profilage d’influenceur de cryptomonnaie pour un tweet anonyme.

Conclusion et perspectives

Les cryptomonnaies ont acquis une importance indéniable dans l'écosystème financier mondial, marquant une révolution dans les méthodes de transaction et d'investissement. Cependant, leur ascension n'est pas sans défis. La volatilité des prix, l'incertitude réglementaire et les préoccupations concernant la sécurité et la fraude pèsent sur leur crédibilité. De plus, l'émergence des médias sociaux a exacerbé ces problèmes en facilitant la diffusion de la désinformation et la manipulation du marché.

Pour relever ces défis, le profilage des influenceurs de cryptomonnaies émerge comme une solution stratégique. Ce processus cherche à mieux comprendre le rôle et l'impact des influenceurs, à détecter les pratiques néfastes et à renforcer la transparence et la confiance sur le marché. Cette approche permet de consolider la confiance et la stabilité du marché des cryptomonnaies tout en protégeant les investisseurs contre les risques potentiels.

L'objectif de notre travail est de concevoir un modèle de profilage d'influenceurs dans le domaine des cryptomonnaies, capable de prédire efficacement l'importance d'un influenceur dans ce domaine en fonction de leur niveau d'influence, de leurs intérêts et de leurs intentions, en utilisant l'apprentissage par transfert. Nous avons présenté deux modèles distincts pour résoudre ce problème : SetFit, une méthode innovante que nous avons été les premiers à utiliser dans la tâche de profilage d'influenceurs de la cryptomonnaie, et DistilBERT, déjà employé dans cette tâche. Ces modèles ont été évalués à l'aide des données d'un ensemble de données Twitter PAN 2023, est divisé en trois sous-tâches distinctes, avec des ressources limitées. Nous avons mené plusieurs expérimentations afin d'évaluer les performances de nos modèles dans différents scénarios et paramètres. Nous avons obtenu des performances satisfaisantes avec le modèle DistilBERT, affichant une F1-score de 80,77%, tandis que le modèle SetFit a atteint une valeur de 82,69%.

Ces résultats relativement acceptables pourraient être attribués à la complexité inhérente du problème de profilage des influenceurs de cryptomonnaie ainsi qu'aux ressources limitées de notre ensemble de données.

Nos perspectives futures dans le domaine du profilage des influenceurs de cryptomonnaie incluent la recherche d'une meilleure précision en explorant diverses approches susceptibles d'améliorer les performances. Nous envisageons d'étudier des ensembles de données plus vastes et variés, au-delà de Twitter, en incluant d'autres plateformes de médias sociaux. Une piste de recherche intéressante serait d'étendre notre approche à d'autres langues, notamment l'arabe, connue pour ses caractéristiques riches et distinctives. Nous prévoyons également d'utiliser des méthodes d'augmentation de données, telles que la génération de nouvelles données synthétiques à partir des données existantes, la traduction de textes dans différentes langues, la création de paraphrases, et l'ajout de bruit contrôlé aux données textuelles. En suivant ces orientations, notre objectif est de fournir des contributions significatives dans le domaine du profilage des influenceurs de cryptomonnaie, visant une compréhension plus approfondie et une meilleure évaluation de leur impact et de leur importance.

Bibliographie

- Abbasi, A., & Chen, H. (2005). Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20(5), 67–75.
- Bernier, S. (2023). *Techniques d'apprentissage machine pour l'estimation du risque suicidaire sur les réseaux sociaux* (Thèse de doctorat non publiée). Université du Québec en Outaouais.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., . . . Amodei, D. (2020). Language models are few-shot learners. *CoRR*, abs/2005.14165.
- Carrasco, S. S., & Rosillo, R. C. (2022). Loscalis at politices 2022 : Political author profiling using beto and maria. In *Proceedings of the iberian languages evaluation forum (iberlef 2022)*. *ceur workshop proceedings, ceur-ws, a coruna, spain*.
- Chinea-Rios, M., Borrego-Obrador, I., Franco-Salvador, M., Rangel, F., & Rosso, P. (2023). Profiling cryptocurrency influencers with few-shot learning. *Overview for PAN Lab at CLEF*.
- Daelemans, W., Kestemont, M., Manjavacas, E., Potthast, M., Rangel, F., Rosso, P., . . . Zangerle, E. (2019, Sep). Overview of pan 2019 : Author profiling, celebrity profiling, cross-domain authorship attribution and style change detection. In F. Crestani et al. (Eds.), *Proceedings of the tenth international conference of the clef association (clef 2019)*. Springer.
- Das, A., Raychawdhary, N., Dozier, G., & Seals, C. D. (2022). Irony and stereotype spreading author profiling on twitter using machine learning : A bert-tfidf based approach.
- Elmanarelbouanani, S., & Kassou, I. (2014). Authorship analysis studies : A survey. *International Journal of Computer Applications*, 86, 22–29.
- Espinosa, D., & Sidorov, G. (2023). Using bert to profiling cryptocurrency influencers. *Working Notes of CLEF*.
- Face, H. (2023a). *Distilbert model card*. Consulté sur <https://huggingface.co/distilbert/distilbert-base-uncased> (Accessed : 2024-05-20)
- Face, H. (2023b). *Tokenizers*. <https://huggingface.co/learn/nlp-course/chapter2/>

4?fw=pt. (NLP Course)

- Ferri-Molla, I., & Santamaria-Jorda, J. (2023). Profiling cryptocurrency influencers with few-shot learning. *Working Notes of CLEF*.
- Girish, K., Hegdev, A., Balouchzahi, F., & Lakshmaiah, S. (2023). Profiling cryptocurrency influencers with sentence transformers. *Working Notes of CLEF*.
- given i=F, f., given=Francisco, given i=M, f.-R., given=Mara, given i=M, f.-S., given=Marc, & given i=P, f., given=Paolo. (s. d.). *Pan23 profiling cryptocurrency influencers with few-shot learning*. Consulté sur <https://zenodo.org/record/7701748> doi: 10.5281/zenodo.7701748
- Joo, Y., Hwang, I., Cappellato, L., Ferro, N., Losada, D., & Müller, H. (2019). Author profiling on social media : An ensemble learning model using various features. *Notebook for PAN at CLEF, 2380*.
- Juola, P., et al. (2008). Authorship attribution. *Foundations and Trends® in Information Retrieval, 1(3)*, 233–334.
- Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing : An introduction to natural language processing, computational linguistics, and speech recognition*.
- Kaur, S. e. K. H., Ravneet et Singh. (2019). Analyse de la paternité du contenu des médias sociaux en ligne. In *Actes de la 2e conférence internationale sur la communication, l'informatique et les réseaux* (pp. 539–549).
- Kowsari, K., Heidarysafa, M., Odukoya, T., Potter, P., Barnes, L. E., & Brown, D. E. (2021). Gender detection on social networks using ensemble deep learning. In *Proceedings of the future technologies conference (ftc) 2020, volume 1* (pp. 346–358).
- M, H., & M.N, S. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process, 5*, 01–11.
- Mendenhall, T. C. (1887). The characteristic curves of composition. *Science, ns-9(214s)*, 237–246.
- Mosteller, F., & Wallace, D. L. (1963). Inference in an authorship problem. *Journal of the American Statistical Association, 58(302)*, 275–309.
- Muslihuddeen, H., Sathvika, P., Sankar, S., Ostwal, S., & Kumar, D. (2023). Profiling cryptocurrency influencers using few-shot learning. *Working Notes of CLEF*.
- Palomino-Garibay, A., Camacho-Gonzalez, A. T., Fierro-Villaneda, R. A., Hernandez-Farias, I., Buscaldi, D., Meza-Ruiz, I. V., et al. (2015). A random forest approach for authorship profiling. In *Proceedings of clef*.
- Para, U., & Patel, M. S. (2023, may). A new term representation method for gender and age prediction. *International Journal on Recent and Innovation Trends in Computing and Communication, 11(5s)*, 90–104. Consulté sur <https://doi.org/10.17762/2Fijritcc.v11i5s.6633> doi: 10.17762/ijritcc.v11i5s.6633
- Radha, D., & Chandra Sekhar, P. (2022). A feature selection technique-based approach for author profiling. In *Intelligent systems and sustainable computing : Proceedings*

- of *icissc 2021* (pp. 583–591). Springer.
- Reimers, N., & Gurevych, I. (2019). Sentence-bert : Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv :1908.10084*.
- Roy, N. (2019). Authorship analysis as a text classification or clustering problem.
- Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert : smaller, faster, cheaper and lighter. *arXiv preprint arXiv :1910.01108*.
- Sentence Transformers. (2024). *paraphrase-mpnet-base-v2 (revision e6981e5)*. Hugging Face. Consulté sur <https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2> doi: 10.57967/hf/2004
- Siino, M., Tesconi, M., Tinnirello, I., et al. (2023). Profiling cryptocurrency influencers with few-shot learning using data augmentation and electra. *Working Notes of CLEF*.
- Siino, M., & Tinnirello, I. (2023). Xlnet with data augmentation to profile cryptocurrency influencers. *Working Notes of CLEF*.
- Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3), 538–556.
- Stamatatos, E. (2016). Authorship verification : A review of recent advances. *Research in Computing Science*, 123, 9–25.
- Statista. (2023). *Internet and social media users in the world 2023 | Statista — statista.com*. <https://www.statista.com/statistics/617136/digital-population-worldwide/>. ([Accessed 30-01-2024])
- Tunstall, L., Reimers, N., Jo, U. E. S., Bates, L., Korat, D., Wasserblat, M., & Pereg, O. (2022). Efficient few-shot learning without prompts. *arXiv preprint arXiv :2209.11055*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Yule, G. U. (1939). On sentence-length as a statistical characteristic of style in prose : With application to two cases of disputed authorship. *Biometrika*, 30, 363–390.
- Zipf, G. K. (1932). *Selected studies of the principle of relative frequency in language*. Cambridge, MA and London, England : Harvard University Press.