TextMine '25

Atelier sur la Fouille de Textes

Organisateurs :

Pascal Cuxac (INIST - CNRS),
Cédric Lopez (Emvista),
Adrien Guille (Univ. Lyon 2)

Organisé conjointement à la conférence EGC
(Extraction et Gestion des Connaissances)
le 28 janvier 2025 à Strasbourg

Editeurs :

Pascal Cuxac - INIST - CNRS
2 rue Jean Zay, CS 10310, 54519 Vandoeuvre les Nancy Cedex
Email : pascal.cuxac@inist.fr

Cédric Lopez - Emvista
Espace Bocaud, 42 rue de la Pierre Plantée, 34830 Jacou
Email : cedric.lopez@emvista.com

Publisher:

Pascal Cuxac, Cédric Lopez
2 rue Jean Zay
54519 Vandoeuvre les Nancy Cedex

Vandoeuvre les Nancy Cedex, France, 2024

PRÉFACE

C'est une évidence que de dire que nous sommes entrés dans une ère où la donnée textuelle sous toute ses formes submerge chacun de nous que ce soit dans son environnement personnel ou professionnel : l'augmentation croissante de documents nécessaires aux entreprises ou aux administrations, la profusion de données textuelles disponibles via Internet, le développement des données en libre accès (OpenData), les bibliothèques et archives en lignes, les medias sociaux ne sont que quelques exemples illustrant l'évolution de la notion de texte, sa diversité et sa prolifération.

Face à cela les méthodes automatiques de fouille de données (data mining), et plus spécifiquement celles de fouille de textes (text mining) sont devenues incontournables. Récemment, les méthodes de deep learning ont créées de nouvelles possibilités de recherche pour traiter des données massives et de grandes dimensions. Cependant, de nombreuses questions restent en suspens, par exemple en ce qui concerne la gestion de gros corpus textuels multi-thématiques. Pouvoir disposer d'outils d'analyse textuelle efficaces, capables de s'adapter à de gros volumes de données, souvent de nature hétérogène, rarement structurés, dans des langues variées, des domaines très spécialisés ou au contraire de l'ordre du langage naturel reste un challenge.

La fouille de textes couvre de multiples domaines comme, le traitement automatique des langues, l'intelligence artificielle, la linguistique, les statistiques, l'informatique et les applications sont très diversifiées, que ce soit la recherche d'information, le filtrage de spam, le marketing, la veille scientifique ou économique, la lutte antiterroriste...

Le but de cet atelier est de réunir des chercheurs sur la thématique large de la fouille de textes. Cet atelier vise à offrir une occasion de rencontres pour les universitaires et les industriels, appartenant aux différentes communautés de l'intelligence artificielle, l'apprentissage automatique, le traitement automatique des langues, pour discuter des méthodes de fouille de textes au sens large et de leurs applications.

P. Cuxac — INIST-CNRS
C. Lopez — Emvista
A. Guille — Université Lyon 2

Membres du comité de lecture

Le Comité de Lecture est constitué de:

Pauline Armary (Université de Bourgogne, Dijon)

Hugo Attali (LIPN, Paris)

Pascal Cuxac (INIST-CNRS, Nancy)

Emma Effa-Bella (Sorbonne Université, Paris)

Adrien Guille (ERIC, Univ. Lyon2, Lyon)

Nicolas Gutehrlé (Université Franche-Comté, Besançon)

Vincent Lemaire (Orange Labs, Lannion)

Cédric Lopez (Emvista, Montpellier)

Jean-Luc Minel (MoDyCo, Université Paris Ouest Nanterre, Paris)

Maxime Prieur (Airbus Defense and Space, Paris)

Solen Quiniou (LS2N, Nantes Université, Nantes)

Christophe Rodrigues (DVRC, Université Léonard de Vinci, Paris)

Sylvain Verdy (Emvista, Montpellier)

TABLE DES MATIÈRES

Session Exposés Invités

Session Exposés

Est ce que les LLM véhiculent des stéréotypes lors de la détection de position ?

Christine Largeron

Université Jean Monnet à Saint-Etienne ; Laboratoire Hubert Curien
Christine.Largeron@univ-st-etienne.fr

Résumé :

Les grands modèles linguistiques héritent des stéréotypes de leurs données de pré-entraînement, ce qui conduit à un comportement biaisé envers certains groupes sociaux dans de nombreuses tâches de traitement du langage naturel, telles que la détection des discours haineux ou l'analyse des sentiments. Étonnamment, l'évaluation de ce type de biais dans les méthodes de détection de position a été largement négligée par la communauté. La détection de position consiste à étiqueter une déclaration comme étant, contre, en faveur ou neutre envers une cible spécifique et fait partie des tâches de NLP les plus sensibles, car elle est souvent liée aux tendances politiques.

Dans cette présentation, nous étudierons les biais des grands modèles linguistiques lors de la détection de position afin de voir s'ils véhiculent ou non des stéréotypes.

Extraction d’information dans un contexte spécifique

Nihel Kooli

Agence ministérielle de l'intelligence artificielle de défense
nihel.kooli@intradef.gouv.fr

Résumé :

Malgré les avancées notables dans le domaine d’extraction d’informations, celui-ci présente encore des défis notamment quand il s’agit d’étudier des contextes spécifiques. Ces derniers peuvent être liés à un domaine de spécialité, à une langue autre que l’anglais (langues moins dotées) ou à un type de documents présentant un langage spécifique (ex : textes issus des réseaux sociaux).

Durant ce talk, nous nous intéresserons aux tâches de reconnaissance d’entités et de détection d’évènements dans des contextes métiers spécifiques. Nous ferons également un focus sur les challenges en lien avec l’élaboration de corpus pour l’apprentissage et l’évaluation de ces tâches.

Adaptation d'un modèle de langue encodeur-décodeur pour l'extraction de relations dans des rapports de renseignement

Adrien Guille

ERIC Lyon 2, EA 3083, Université de Lyon
https://github.com/adrienguille/textmine2025

1 Introduction

Cette proposition au Défi TextMine 2025 (Prieur et al., 2025) s'inscrit dans le cadre moderne du traitement automatique de la langue, en définissant la tâche d'extraction de relations comme une tâche de génération de texte.

Formulation de la tâche On s'inspire de la démarche proposée par Zhang et al. (2023), où l'extraction de relations est formulée comme un questionnaire à choix multiples, un modèle de langue se chargeant de choisir les options susceptibles de correspondre à des relations présentes dans un texte. Plutôt que d'inclure toutes les relations candidates en un seul prompt par le biais d'un QCM, on choisit de présenter chaque relation candidate dans un prompt indépendant, sous la forme d'une question fermée. Les relations candidates sont identifiées suivant les motifs (*type d'entité source*, *type de relation*, *type d'entité cible*) observés dans les données d'entraînement. Ci-après, un exemple de prompt et la réponse attendue :

Entrée reçue et sortie attendue

Does the relation (head_entity : [Constance Dupuis], relation_type : is_in_contact_with}, tail_entity : [Airîle, compagnie aérienne]), exists in the following text : "L'avion NY8 de la compagnie Airîle a lancé sa dernière position via le signal radio avant de se crasher dans une forêt en Malaisie le 19 février 2003. La compagnie aérienne a alerté les secours pour évacuer les passagers. Les hélicoptères d'urgence ont retrouvé l'appareil en feu. Les autorités malaisiennes ont recensé 15 morts au total. Cet incident n'a fait que peu de survivants, dont Constance Dupuis, présidente de l'association « des médicaments pour tous » en Grèce. D'après son témoignage, le NY8 a connu une défaillance technique que les pilotes n'ont pas pu contrôler. Les corps ont été transportés par brancard à la morgue." ?

no

Choix du type de modèle de langue Dans la lignée de nos propres travaux (Charpentier et al., 2024) et de travaux connexes tels que ceux de Qorib et al. (2024), qui mettent en avant les capacités sémantiques supérieures des encodeurs par rapport aux décodeurs, nous écartons les modèles de langue basés uniquement sur un décodeur (*e.g.* Llama, Mistral) au profit de modèles de langue basés sur une architecture encodeur-décodeur.

2 Modèles de langues

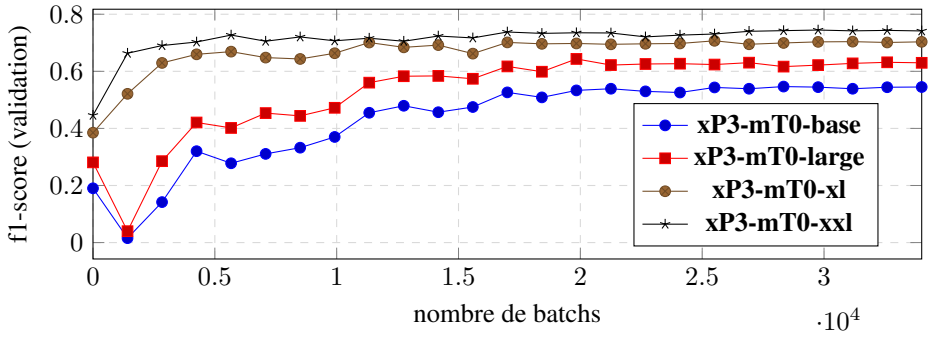
Modèles pré-entraînés On se limite à l'architecture encodeur-décodeur basée sur le bloc Transformer, telle que décrite par Raffel et al. (2020). Plus précisément, on considère le modèle pré-entraîné T5 et des modèles qui en découlent, brièvement décrits ci-après :

— **T5** (Raffel et al., 2020) : Modèle pré-entraîné sur le corpus anglophone C4 avec un objectif de débruitage de passages de texte ;
— **FLAN-T5** Chung et al. (2022) : Modèle **T5** spécialisé pour répondre à des instructions en poursuivant l'entraînement sur le corpus FLAN (où les instructions sont rédigées exclusivement en anglais et les réponses multilingues) ;
— **mT5** (Xue et al., 2021) : Même architecture que **T5** avec un vocabulaire de 250K tokens (au lieu de 32K pour T5), pré-entraîné sur le corpus multilingue mC4 ;
— **xP3-mT0** (Muennighoff et al., 2023) : Modèle **mT5** spécialisé pour répondre à des instructions en poursuivant l'entraînement sur le corpus xP3 (où les instructions sont rédigées exclusivement en anglais et les réponses multilingues).

Adaptation des modèles Pour adapter efficacement les modèles pré-entraînés à la tâche d'extraction de relations dans des rapports de renseignement, on procède à une adaptation de rang faible selon la méthode LoRA (Hu et al., 2022). Pour une matrice $W \in \mathbb{R}^{d_{\text{entrée}} \times d_{\text{sortie}}}$ paramétrant le modèle, cela consiste à approcher les ajustements à lui apporter, $\Delta W \in \mathbb{R}^{d_{\text{entrée}} \times d_{\text{sortie}}}$, par un produit de matrices de rang faible, $A \in \mathbb{R}^{d_{\text{entrée}} \times \text{rang}}$ et $B \in \mathbb{R}^{\text{rang} \times d_{\text{sortie}}}$: $\Delta W \approx A \times B$.

3 Résultats

On répartit les 800 documents annotés en 700 documents utilisés pour l'adaptation (matrices des projections en requêtes et en valeurs uniquement, rang 64) et 100 documents utilisés pour la validation. Par manque de place, on ne présente qu'une expérience sur l'effet de la taille du modèle **xP3-mT0**, sa variante **xxl** se classant première du défi. On observe que la performance est corrélée à la taille du modèle, que ce soit avant ou après l'adaptation. On remarque dans la figure ci-après que les grands modèles nécessitent peu de données pour atteindre leur meilleur score, tandis que l'adaptation des petits modèles est instable. On note par ailleurs que les variantes de **FLAN-T5** atteignent des scores similaires en validation, alors que les scores en test sont nettement inférieurs, ce qui suggère que l'entraînement multilingue de **xP3-mT0** lui confère une meilleure capacité de généralisation en français.



Références

Prieur, M., G. Gadek, A. Guille, H. Rawsthorne, P. Cuxac, et C. Lopez (2025). Défi TextMine'25 – Extraction de relations pour analyser des rapports de renseignement. In *Atelier TextMine (TextMine @ EGC 2025)*.

Charpentier, F., J. Cugliari, et A. Guille (2024). Exploring semantics in pretrained language model attention. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM @ ACL 2024)*.

Qorib, M., G. Moon, et H. T. Ng (2024). Are decoder-only language models better than encoder-only language models in understanding word meaning ? In *Findings of the Association for Computational Linguistics (ACL 2024)*.

Muennighoff, N., T. Wang, L. Sutawika, A. Roberts, S. Biderman, T. Le Scao, M. S. Bari, S. Shen, Z. X. Yong, H. Schoelkopf, X. Tang, D. Radev, A. F. Aji, K. Almubarak, S. Albanie, Z. Alyafeai, A. Webson, E. Raff, et C. Raffel (2023). Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*.

Zhang, K., B. Jimenez Gutierrez, et Y. Su (2023). Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics (ACL 2023)*.

Chung, H. W., L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, D. Valter, S. Narang, G. Mishra, A. W. Yu, V. Zhao, Y. Huang, A. M. Dai, H. Yu, S. Petrov, E. H. hsin Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, et J. Wei (2022). Scaling instruction-finetuned language models. *Journal of Machine Learning Research 25*.

Hu, E. J., yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, et W. Chen (2022). LoRA : Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR 2022)*.

Xue, L., N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, et C. Raffel (2021). mT5 : A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL 2021)*.

Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, et P. J. Liu (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research 21*.

ICB@Défi TextMine'25 : Extraction de relations pour l'analyse des rapports de renseignement

Hussam Ghanem*, Daren Hacbekri*
Christophe Cruz*

*ICB, UMR 6306, CNRS, Université de Bourgogne, 21000 Dijon, France
https://icb.u-bourgogne.fr/

1 Introduction

Le défi TextMine'25 (Prieur et al., 2025), proposé par Airbus Defence and Space, se concentre sur l'extraction automatique des relations entre entités dans des rapports de renseignement. Cette tâche, cruciale pour l'analyse dans les domaines de la défense et du renseignement, reste un défi scientifique nécessitant souvent une intervention humaine. Le défi vise à comparer des solutions technologiques publiques et privées à l'aide de données textuelles fictives annotées, issues du projet POPCORN (Giordano, 2024). Les participants doivent identifier et annoter des relations parmi 37 catégories, telles que LOCATED_IN ou PART_OF. Ces données, simulant des scénarios réels, permettent de tester les capacités des modèles à comprendre les interactions complexes entre acteurs, événements et attributs, avec des implications directes pour l'analyse de rapports de renseignement.

Le jeu de données TextMine'25 comprend 800 documents factices annotés pour l'entrainement et 400 documents pour l'évaluation, organisés en fichiers CSV : **train.csv** (textes avec entités et relations), **test.csv** (textes avec entités uniquement) et **sample_submission.csv** (exemple de soumission). Les entités sont représentées par des dictionnaires (id, mentions, type), tandis que les relations dans l'ensemble d'entraînement sont des triplets (id source, type de relation, id cible). Le corpus inclut 38 types de relations, avec plusieurs types d'identités et d'attributs. L'évaluation utilise le score Macro F1, calculé comme la moyenne des scores F1 pour chaque type de relation. L'ensemble des informations sont disponibles sur le site Kaggle [1]

2 Prompt Engineering

Nous avons abordé le défi TextMine'25 [2] en exploitant le **Prompt Engineering** pour extraire des relations à partir de textes en utilisant le grand modèle de langage (LLM) **Gemini-1.5-pro-002** [3]. En définissant des instructions explicites et en normalisant les formats de sor-

1. Défi TextMine 2025 : https://www.kaggle.com/competitions/defi-text-mine-2025/
2. Notre code sur Github : https://github.com/ChristopheCruz/2024_Kaggle_competition
3. Modèles Gemini : https://ai.google.dev/gemini-api/docs/models/gemini?hl=fr#gemini-1.5-flash

tie, nous avons structuré les relations sous forme de triplets : `[Subject_id, Relation Type, Object_id]`.

Notre approche combine le **zero-shot prompting (ZSP)** (Caufield et al., 2023) et le **few-shot prompting (FSP)** (Han et al., 2023). Le ZSP correspond à des prompts formulés sans fournir d'exemples d'entrée-sortie, tandis que le FSP implique l'utilisation de prompts enrichis avec des exemples illustratifs. Dans le prompt, le texte et les entités (ainsi que leurs attributs) sont fournis comme entrée, et le modèle est invité à extraire les relations correspondantes. Une hiérarchie stricte a été adoptée pour structurer les entités et attributs, renforçant la cohérence des relations extraites. Des règles ont également été imposées, telles que la priorisation des entités spécifiques (enfants) et un formatage rigoureux des résultats. Par exemple, la relation `IS_BORN_IN` était limitée aux types `Person` (sujet) et `Place` (objet).

Initialement, le modèle rencontrait des difficultés avec le format et la validité des relations. Par itérations successives, dans notre prompt, nous avons affiné les instructions et mis en place des directives pour éliminer les incohérences, éviter les explications superflues, et garantir une sortie conforme.

Défis et solutions. Au départ, Gemini ajoutait des explications superflues autour des triplets. L'interdiction explicite de tels textes garantissait des résultats plus propres. La restriction des sujets à la liste des entités autorisées a réduit les erreurs d'identification des sujets. Donner la priorité aux entités enfants par rapport aux types parents a amélioré la granularité des relations extraites. Des règles de formatage strictes ont permis de remédier aux incohérences en matière de représentation d'ID incorrecte et de structure de relation.

3 Résultats et conclusion

Les résultats de notre approche, présentés dans le tableau 1, montrent une amélioration significative des performances du modèle Gemini-1.5-pro-002 à mesure que des exemples supplémentaires sont incorporés dans les prompts. En zero-shot, le modèle atteint un Macro F1 de 0.307, ce qui reflète une capacité limitée à extraire correctement les relations sans exemples. L'ajout de 5 exemples structurés (5-shots) entraîne une augmentation notable du score, atteignant 0.457. Cette progression indique que le modèle bénéficie fortement de directives explicites et de contextes démonstratifs.

Cependant, l'ajout de 10 exemples (10-shots) n'entraîne qu'une amélioration marginale, avec un Macro F1 de 0.464. Cette faible différence entre les scénarios 5-shots et 10-shots suggère que l'utilisation de plus d'exemples ne garantit pas nécessairement une amélioration substantielle des performances. Cela pourrait s'expliquer par une saturation des capacités d'apprentissage du modèle dans ce cadre particulier, où un petit nombre d'exemples bien structurés suffit pour atteindre des performances optimales.

Ainsi, nos résultats indiquent qu'une approche concise, exploitant un nombre limité mais pertinent d'exemples, est à la fois efficace et efficiente pour cette tâche spécifique. Cette stratégie réduit également les coûts de traitement tout en maintenant des performances élevées.

	Zero-shot	5-shots	10-shots
Macro F1	0.307	0.457	0.464

TAB. 1 – *Résults with Gemini-1.5-pro-002.*

Références

Prieur, M., G. Gadek, H. Rawsthorne, A. Guille, P. Cuxac, et C. Lopez (2025). Défi textmine'25 -extraction de relations pour analyser des rapports de renseignement. actes de l'atelier textmine'25, p. à paraître, extraction et gestion des connaissances 2025 (egc'25).

Giordano, A., e. a. (2024). Fictional and synthetic intelligence reports for named entity recognition and relation extraction tasks. in proceedings of kes'24, to appear.

Caufield, J. H., H. Hegde, V. Emonet, N. L. Harris, M. P. Joachimiak, N. Matentzoglu, H. Kim, S. A. Moxon, J. T. Reese, M. A. Haendel, et al. (2023). Structured prompt interrogation and recursive extraction of semantics (spires) : A method for populating knowledge bases using zero-shot learning. *arXiv preprint arXiv :2304.02711*.

Han, J., N. Collier, W. Buntine, et E. Shareghi (2023). Pive : Prompting with iterative verification improving graph-based generative capability of llms. *arXiv preprint arXiv :2305.12392*.

Summary

The TextMine'25 challenge, by Airbus Defence and Space, focuses on extracting relationships in intelligence reports. We apply Large language models (Gemini) to identify relations between entities, reducing manual efforts and advancing relation extraction methods on annotated data.

CEA-List@TextMine'25 : ensemble, c'est mieux ?

Arthur Peuvot*, Romaric Besançon*, Olivier Ferret*, Benjamin Labbé*, Clément Maurer*, Nasredine Semmar*, Sondes Souihi*

*Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
{nom.prenom}@cea.fr,
https://kalisteo.cea.fr/

Résumé. Dans cet article, nous présentons une contribution du CEA-List au défi TextMine'25 se focalisant sur deux principales extensions du modèle ATLOP, l'une axée sur la prise en compte des corrélations entre relations, l'autre sur l'exploitation de la notion de preuve. Nous les associons au travers de plusieurs configurations d'ensemble et comparons les combinaisons ainsi obtenues à ces deux extensions. Le code correspondant à ces expérimentations est disponible à l'adresse suivante : https://github.com/CEA-LIST/textmine-2025.

1 Introduction

Si l'extraction de relations intraphrastiques entre entités a fait l'objet d'un volume de travaux historiquement très important, la transposition de cette extraction à l'échelle du document est plus récente, mais en développement comme en attestent (Delaunay et al., 2023; Zheng et al., 2023), en particulier depuis la mise à disposition de corpus annotés tels que DocRED (Yao et al., 2019). Cette transposition se heurte à une difficulté principale : en sortant du cadre de la phrase, les processus d'extraction se privent de la possibilité de s'appuyer de façon systématique sur les informations linguistiques disponibles à l'échelle phrastique, comme les relations syntaxiques. En contrepartie, ils gagnent la possibilité de collecter et d'agréger à l'échelle du document des informations à propos des entités impliquées dans les relations, de même que des informations à propos des relations qu'elles entretiennent. Ces entités peuvent en effet avoir plusieurs occurrences au sein d'un même document. Chacune de ces occurrences s'inscrit dans un contexte permettant d'enrichir la représentation de l'entité correspondante. Cet enrichissement inclut également les relations que l'entité en question entretient avec d'autres entités. Enfin, il est possible qu'une même relation ait plusieurs occurrences dans un même document. Dès lors, un défi important de l'extraction de relations au niveau des documents réside dans la capacité des modèles à allier la précision que représente le niveau des occurrences et la richesse apportée par l'agrégation des informations à l'échelle du document concernant les éléments constitutifs des relations. Selon les cas, les modèles mettent un accent plus ou moins fort sur l'une ou l'autre de ces dimensions.

Les modèles à base de graphe, incarnés typiquement par (Christopoulou et al., 2019), adoptent ainsi le partie pris de construire des représentations au niveau des occurrences et de prendre en compte les informations liées à ces occurrences par le biais d'un GNN (Graph Neural Network) exploitant une structure de graphe reliant les différentes occurrences d'entités. À

l'inverse, un modèle comme ATLOP (Zhou et al., 2021) s'appuie essentiellement sur l'agrégation des représentations des différentes occurrences des entités impliquées dans les relations candidates. Cette dimension globale s'étend aux corrélations entre relations avec LACE (Huang et Lin, 2024), une variante d'ATLOP. Néanmoins, le niveau global et le niveau des occurrences peuvent aussi être associés. Les modèles EIDER (Xie et al., 2022) et DREEAM (Ma et al., 2023), fondés aussi sur ATLOP, réintroduisent ainsi le niveau des occurrences en proposant et utilisant la notion de preuve, que l'on peut voir comme des instances de relations.

2 Modèles de référence

Parmi les travaux notables réalisés ces dernières années dans le domaine de l'extraction de relations entre entités à l'échelle du document, le modèle ATLOP, qui sert de base à de nombreux modèles, a retenu notre attention. Reposant sur un encodeur BERT, ATLOP modélise chaque relation candidate par le couple de ses entités. La représentation de chacune des entités résulte de l'agrégation par le pooling *logsumexp* [1] des représentations de ses différentes mentions. Le modèle introduit également un seuil adaptatif appris comme une classe supplémentaire, éliminant ainsi la nécessité de fixer un seuil de décision. Enfin, ATLOP exploite les matrices d'attention de BERT pour identifier les informations contextuelles propres à chaque paire d'entités. Dans le cadre de l'atelier TextMine'25 (Prieur et al., 2025), nous avons choisi d'explorer trois modèles fondés sur ATLOP, à partir desquels nous avons réalisé différentes expériences.

LACE La première extension d'ATLOP que nous avons exploitée est le modèle LACE, qui repose sur une architecture composée de trois modules : un module d'encodage, un module de corrélation des relations et un module de classification. Son module d'encodage repose sur les principes d'ATLOP pour produire les représentations contextuelles des mentions d'entités, notamment grâce au pooling *logsumexp*. Le module de corrélation des relations modélise les interdépendances entre relations sous forme de probabilités conditionnelles. Ces informations sont capturées dans une matrice de corrélation fondée sur les cooccurrences observées dans les données d'entraînement. Pour éviter le sur-apprentissage, cette matrice est filtrée à l'aide d'un seuil. LACE exploite ensuite un Graph Attention Network (GAT) (Veličković et al., 2018), doté d'un mécanisme d'attention multi-têtes, pour agréger les représentations des relations. Le GAT permet de réduire le problème de sur-lissage (over-smoothing) des réseaux de graphes, où les représentations des nœuds, ici correspondant aux relations, dans un graphe convergent vers des valeurs trop similaires. Enfin, tout comme ATLOP, le module de classification s'appuie sur une couche bilinéaire pour prédire les relations en combinant les représentations des entités et des relations. Cependant, ce module introduit une version révisée de la fonction de perte AT (Adaptive Thresholding) d'ATLOP, appelée perte MAT (Multi-relation Adaptive Thresholding), permettant de mieux gérer les cas de classification multi-étiquette.

EIDER Alors qu'ATLOP et LACE privilégient des informations agrégées, le modèle EIDER, reprenant lui aussi les mécanismes de base d'ATLOP, propose d'exploiter des *preuves*, définies comme les phrases d'un document permettant d'attester la présence d'une relation. EIDER

1. Ce pooling s'apparente à une forme de max pooling.

combine grâce à un apprentissage multi-tâche l'extraction des relations et celle des preuves. Chaque tâche dispose de son propre classifieur mais elles partagent le même encodeur. Les modules d'encodage et d'extraction des relations utilisent les mêmes systèmes que ceux d'ATLOP, à savoir le pooling *logsumexp*, la perte de seuil adaptatif et le pooling de contexte localisé. Le module d'extraction des preuves permet d'identifier des phrases pertinentes, afin qu'elles puissent être utilisées pour renforcer la prédiction des relations. Dans le cas où les annotations de preuves ne sont pas disponibles, EIDER propose des règles heuristiques pour générer des pseudo-étiquettes. L'entraînement du modèle se fait de manière conjointe, en combinant la perte associée à l'extraction des relations et la perte pour l'extraction des preuves. Enfin, en phase d'inférence, EIDER fusionne les prédictions effectuées sur le document originel et sur les preuves extraites. L'idée ici est que, même si les preuves extraites peuvent ne pas être complètes, leur fusion avec les informations du document originel permet de compenser les lacunes possibles dans l'extraction des preuves. Cette fusion se fait à l'aide d'une couche d'assemblage, qui combine les scores de prédiction des deux sources.

DREEAM Enfin, nous avons considéré le modèle DREEAM qui, tout comme EIDER, propose d'utiliser les preuves pour améliorer les performances de la tâche d'extraction de relations. DREEAM attribue des poids plus élevés aux tokens correspondant aux preuves qu'aux autres, permettant ainsi de concentrer les mécanismes d'attention sur les preuves. De plus, pour surmonter le manque d'annotations de preuves, DREEAM propose une stratégie de supervision distante. Pour cela, un modèle enseignant, préalablement entraîné sur des données annotées manuellement, est utilisé pour identifier des pseudo-preuves dans les données annotées automatiquement. Ces dernières sont ensuite utilisées pour entraîner un modèle étudiant à effectuer simultanément les tâches d'extraction de preuves et de relations.

3 Études réalisées

Les trois modèles présentés ci-dessus ont servi de base à toutes les études que nous avons menées, qui se sont focalisées sur les quatre points suivants :

— les différentes façons de définir des preuves ;
— la représentation des entités d'une relation ;
— l'impact du type de classification : mono vs. multi-étiquette ;
— l'intérêt et les moyens de prendre en compte les dépendances entre les types de relations.

Preuves Concernant la construction des preuves, les stratégies de construction heuristique d'EIDER et la prédiction par entraînement supervisé distant de DREEAM ont été généralisées de façon croisée à ces deux modèles. Pour la seconde méthode, un modèle enseignant, entraîné sur le corpus annoté DocRED (Yao et al., 2019), a permis de générer les pseudo-preuves pour le jeu de données TextMine'25. Pour cela, la couche bilinéaire de classification du modèle a été redéfinie afin d'avoir une couche adaptée au nombre de classes du jeu de données TextMine'25. L'intuition ici est que les connaissances acquises par le modèle peuvent être transférées à de nouveaux types d'entités. Des évaluations ont été faites grâce aux prédictions du modèle enseignant et grâce aux règles heuristiques. Nous avons également combiné ces deux approches en appliquant les règles heuristiques si aucune preuve n'était prédite par le modèle enseignant.

Mono vs. multi-étiquette L'extraction des relations à l'échelle d'un document est intrinsèquement un problème de classification multi-étiquette dans la mesure où plusieurs types de relations peuvent intervenir pour une même paire d'entités. Néanmoins, il s'agit d'un cas globalement peu fréquent et comparée à une classification multiclasse, une classification multi-étiquette nécessite de définir un seuil pour décider de retenir ou non une étiquette, ce qui constitue une difficulté supplémentaire. Nous avons donc testé l'importance de la capacité d'un modèle à prédire plusieurs types de relations pour un couple d'entités donné, en prenant LACE comme cible. Dans le cas mono-étiquette ($\text{LACE}_{\text{mono}}$), uniquement la relation ayant le plus haut score est sélectionnée comme prédiction alors que dans le cas multi-étiquette ($\text{LACE}_{\text{multi}}$), le seuil adaptatif d'ATLOP est utilisé.

Représentation des entités Pour créer la représentation contextuelle d'un couple d'entités, ATLOP utilise uniquement les représentations et les attentions de la dernière couche. Or, des travaux tels que (Tuo et al., 2022) ont montré qu'utiliser la moyenne des représentations ou des attentions de plusieurs couches pouvait améliorer les performances. Ainsi, nous avons réalisé avec LACE des expériences testant différentes combinaisons : la moyenne des représentations de toutes les couches ($\text{LACE}_{\text{m\_embeddings}}$), la moyenne des attentions de toutes les couches ($\text{LACE}_{\text{m\_attentions}}$) ou les deux combinées ($\text{LACE}_{\text{m\_embeddings+m\_attentions}}$).

Dépendances entre relations LACE permet d'obtenir des informations sur les interdépendances entre relations grâce à son module de corrélation et apporte une meilleure gestion des cas de classification multi-étiquette grâce à sa fonction de perte MAT. Nous avons ajouté ces outils à l'exploitation des preuves de DREEAM et EIDER, qui permettent de préciser le contexte pour un couple d'entités, en faisant l'hypothèse que preuves et dépendances entre relations sont a priori complémentaires.

Par ailleurs, toujours concernant les dépendances entre relations, pour évaluer l'importance du mécanisme d'attention utilisé par le GAT dans le modèle LACE, nous avons remplacé ce GAT par un GCN (Graph Convolutional Network). Pour agréger les représentations des relations, un GCN utilise des opérations de convolution alors qu'un GAT réalise des agrégations pondérées grâce à l'attention portée aux relations voisines. L'objectif est de comprendre si l'utilisation d'un GAT permet de mieux capturer les interdépendances entre relations qu'un autre type de GNN.

4 Expérimentations

4.1 Cadre des expérimentations

Pour mener nos expérimentations, nous avons divisé les données d'apprentissage du défi TextMine'25 en trois sous-ensembles : 80 % pour l'entraînement, 10 % pour la validation et 10 % pour le test. La séparation a été faite de sorte que ces proportions soient respectées au mieux pour chaque type de relation. Tous les résultats présentés ci-dessous sont issus de moyennes effectuées sur 10 entraînements afin d'assurer la fiabilité des résultats.

Par ailleurs, toutes les expériences présentées ici ont été réalisées dans des conditions identiques afin d'assurer la reproductibilité et la comparabilité des résultats. Ainsi, pour chaque expérience, une graine est fixée pour assurer le contrôle des opérations comportant une part

d'aléatoire : initialisation des poids des couches ajoutées à l'encodeur, mécanisme de dropout et création des batchs d'entraînement. Nous avons également utilisé un même ensemble d'hyperparamètres pour nos différents modèles, sachant que ceux-ci ont le même soubassement. Ces hyperparamètres se définissent ainsi :

- l'encodeur `xlm-roberta-large` comme modèle de base, dont le caractère multilingue lui permet de mieux prendre en compte les données en français du défi TextMine'25 ;
- taux d'apprentissage :
  - la valeur du taux d'apprentissage a été fixée à 3×10^{-5} pour toutes les couches de l'encodeur ;
  - un taux d'apprentissage spécifique fixé à 1×10^{-4} a été attribué aux couches ajoutées à l'encodeur pour construire les modèles ;
- un warmup ratio de 0,06 ;
- l'optimiseur Adam avec un epsilon fixé à 1×10^{-6} ;
- 30 époques d'entraînement.

4.2 Résultats des expérimentations

Nous rendons compte ici des résultats des expérimentations concernant les points d'étude introduits à la section 3.

Preuves Le tableau 1 présente les performances obtenues par EIDER et DREEAM en fonction de la méthode utilisée pour construire les preuves. L'utilisation d'un modèle enseignant pour prédire les preuves permet d'atteindre un niveau de résultat légèrement supérieur à l'utilisation de règles heuristiques. Cependant, cette approche nécessite davantage de ressources, notamment pour l'entraînement du modèle et son utilisation lors de la phase d'inférence. Par conséquent, pour les expériences suivantes, nous privilégierons l'utilisation de règles heuristiques, qui offrent un compromis plus avantageux entre simplicité et efficacité.

	EIDER		DREEAM	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Prédictions	**79,45** $\pm 0,45$	$55,58 \pm 2,46$	$79,44 \pm 0,67$	**56,41** $\pm 2,86$
Règles heuristiques	$79,27 \pm 0,62$	$55,03 \pm 2,35$	$79,39 \pm 0,95$	$56,01 \pm 2,78$
Prédictions + règles heuristiques	$79,15 \pm 0,62$	$53,86 \pm 1,95$	79.03 ± 0.94	55.32 ± 3.13

TAB. 1 – *Moyennes et écarts-types des performances des modèles EIDER et DREEAM pour les différentes approches de construction de preuves sur 10 expériences.*

Mono vs. multi-étiquette Le tableau 2 montre que le modèle LACE en configuration multi-étiquette obtient des performances sensiblement supérieures à celles de la configuration mono-étiquette. La capacité du modèle à gérer plusieurs étiquettes simultanément en multi-étiquette permet de façon générale une identification plus précise des relations complexes dans les données. Les résultats du tableau 2 illustrent qu'une telle identification est en pratique nécessaire dans un nombre non négligeable de cas pour les données TextMine'25. Cependant, nous avons

observé que le modèle mono-étiquette peut être avantageux pour certaines relations spécifiques, comme HAS_FOR_WIDTH, HAS_FOR_LENGTH, et HAS_FOR_HEIGHT, dont les entités peuvent être très similaires. Dans ces cas, le traitement mono-étiquette pourrait favoriser une meilleure différenciation de ces relations.

	Micro-F1	Macro-F1
$\text{LACE}_{\text{mono}}$	62,68 $\pm$ 0,55	65,07 $\pm$ 1,7
$\text{LACE}_{\text{multi}}$	**68,43** $\pm$ 0,78	**70,08** $\pm$ 1,53

TAB. 2 – *Moyennes et écarts-types du modèle LACE en mono-étiquette et multi-étiquette sur 10 expériences.*

Les tableaux 1 et 2 mettent par ailleurs en évidence une nette supériorité du modèle LACE par rapport aux modèles EIDER et DREEAM concernant la macro-F1, cette supériorité s'inversant pour la micro-F1. On peut en déduire que LACE obtient des résultats assez similaires pour tous les types de relations tandis que les modèles EIDER et DREEAM, pour leur part très comparables pour ces deux mesures, obtiennent des résultats particulièrement élevés pour les types de relations les plus représentés.

Représentation des entités Les performances du modèle LACE selon différentes configurations de construction des représentations et des attentions associées aux couples d'entités sont présentées dans le tableau 3. Les expériences ont été réalisées dans un contexte multi-étiquette. Ces résultats montrent que l'utilisation de la moyenne des représentations ou des attentions sur toutes les couches ne permet pas d'augmenter les performances par rapport à l'approche de base, qui repose uniquement sur les représentations et les attentions de la dernière couche. Il est possible que la simple moyenne sur toutes les couches dilue des informations spécifiques capturées dans les couches les plus pertinentes. Une piste intéressante serait d'explorer des approches alternatives inspirées de (Tuo et al., 2022), comme l'utilisation des représentations issues d'un sous-ensemble de couches sélectionnées, ou encore la réalisation d'une combinaison linéaire des couches, avec l'apprentissage du poids de chaque couche dans cette combinaison.

	Micro-F1	Macro-F1
LACE	**68,43** $\pm$ 0,78	**70,08** $\pm$ 1,53
$\text{LACE}_{\text{m\_embeddings}}$	67,47 $\pm$ 0,59	60,88 $\pm$ 1,88
$\text{LACE}_{\text{m\_attentions}}$	68,0 $\pm$ 0,63	66,84 $\pm$ 1,79
$\text{LACE}_{\text{m\_embeddings+m\_attentions}}$	67,46 $\pm$ 0,43	62,49 $\pm$ 2,12

TAB. 3 – *Moyennes et écarts-types du modèle LACE pour différentes configurations de construction des représentations et des attentions associés aux couples d'entités, sur 10 expériences.*

Dépendances entre relations Le tableau 4 montre que l'utilisation d'un GCN, au lieu d'un GAT, n'a pas d'impact sur la micro-F1 mais provoque une légère baisse de la macro-F1. Ce constat suggère que le GAT permet de capturer les interdépendances entre les relations de façon plus uniforme.

	Micro-F1	Macro-F1
LACE_{GCN}	68,34 ± 0,49	68,61 ± 1,86
LACE_{GAT}	**68,43** ± 0,78	**70,08** ± 1,53

TAB. 4 – *Moyennes et écarts-types du modèle LACE avec différents GNNs sur 10 expériences.*

Enfin, le tableau 5 montre les performances des modèles EIDER et DREEAM avec et sans l'ajout du module de corrélation des relations de LACE. L'intégration des informations d'interdépendance entre relations que fournit le graphe ne profite finalement pas à ces modèles et une chute des performances est même observée. Cette tendance pourrait résulter d'une complexité trop importante du modèle ou d'un mauvais réglage des hyperparamètres liés au graphe.

	EIDER		DREEAM	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Sans graphe	$79,27 \pm 0,62$	$55,03 \pm 2,35$	**79,39** $\pm 0,95$	**56,01** $\pm 2,78$
Avec graphe	79.1 ± 0.42	53.57 ± 3.82	$78,09 \pm 0,44$	$51,4 \pm 1,84$

TAB. 5 – *Moyennes et écarts-types des performances des modèles EIDER et DREEAM avec et sans graphe, sur 10 expériences.*

5 Méthodes d'ensemble

Après avoir évalué la performance de nos différents modèles et de leurs variantes, nous avons évalué leur association par le biais de méthodes d'ensemble, soit au travers de l'association de plusieurs instances d'un même modèle, soit au travers de l'association de modèles différents. Concernant le premier type d'association, le tableau 6 met en regard la performance moyenne de chaque modèle considéré pour 10 instances, chacune entraînée avec une graine aléatoire différente, et la performance d'un ensemble de 10 instances de ces mêmes modèles, construit par un vote à la majorité absolue au niveau de chaque relation identifiée. La méthode d'ensemble profite à LACE dans les deux configurations, avec de nettes améliorations des scores micro et macro-F1. Au contraire, à l'exception d'une légère augmentation du score de micro-F1 de DREEAM, EIDER et DREEAM ne semblent pas bénéficier du vote majoritaire absolu.

Comme mentionné précédemment, LACE atteint les meilleures performances en macro-F1, ce qui indique une meilleure performance pour les relations rares (contexte few-shot). À l'inverse, EIDER et DREEAM obtiennent de meilleurs scores en micro-F1, ce qui reflète leur aptitude à bien prédire des relations fréquentes. Afin d'exploiter les complémentarités de ces différents types de modèles, nous avons testé leur combinaison via un vote majoritaire entre 10 instances de chaque type de modèle. La configuration mono-étiquette de LACE est également utilisée en raison de l'amélioration notable des performances sur des relations spécifiques telles que HAS_FOR_WIDTH, HAS_FOR_LENGTH, et HAS_FOR_HEIGHT.

	Moyenne sur 10 runs		Ensemble de 10 runs	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
$\text{LACE}_{\text{mono}}$	$62,68 \pm 0,55$	$65,07 \pm 1,70$	63,37	67,3
$\text{LACE}_{\text{multi}}$	$68,43 \pm 0,78$	**70,08** $\pm 1,53$	70,62	**71,36**
EIDER	$79,27 \pm 0,62$	$55,03 \pm 2,35$	78,89	51,16
DREEAM	**79,39** $\pm 0,95$	$56,01 \pm 2,78$	**80,14**	53,42

TAB. 6 – *Moyennes et écarts-types vs. vote majoritaire absolu avec 10 modèles.*

Les résultats des différentes associations ainsi testées, présentés dans le tableau 7, montrent que le mode de combinaison adopté ne permet pas de tirer partie de la complémentarité des modèles. Les seules améliorations observées concernent les combinaisons impliquant $\text{LACE}_{\text{multi}}$ et un autre modèle, EIDER ou DREEAM, et tendent plutôt à accentuer les tendances de ces deux modèles. Le score de micro-F1 atteint ainsi 82, ce qui est supérieur aux 78 et 80 respectivement atteints par les ensembles de EIDER et DREEAM (cf. tableau 6). Cependant, le score de macro-F1 n'atteint pas les performances de $\text{LACE}_{\text{multi}}$.

	Micro-F1	Macro-F1
Combinaison d'ensembles $\text{LACE}_{\text{multi}}$ + $\text{LACE}_{\text{mono}}$	66,15	68,79
Combinaison d'ensembles $\text{LACE}_{\text{multi}}$ + EIDER	82,13	68,19
Combinaison d'ensembles $\text{LACE}_{\text{multi}}$ + DREEAM	**82,32**	65,98
Combinaison d'ensembles EIDER + DREEAM	80,45	52,37
Combinaison d'ensembles $\text{LACE}_{\text{multi}}$ + $\text{LACE}_{\text{mono}}$ + EIDER	73,88	**70,78**
Combinaison d'ensembles $\text{LACE}_{\text{multi}}$ + $\text{LACE}_{\text{mono}}$ + DREEAM	73,91	70,62

TAB. 7 – *Performances des différentes combinaisons de modèles.*

Les méthodes d'ensemble fondées sur le vote majoritaire absolu n'ont pas conduit à des améliorations significatives des performances tout en augmentant considérablement la consommation énergétique en raison de l'entraînement et de l'inférence de plusieurs modèles. Il serait néanmoins intéressant d'explorer d'autres approches ensemblistes, telles que le boosting (Freund et Schapire, 1996) ou le stacking (Smyth et Wolpert, 1997).

6 Soumissions

Pour les soumissions finales du défi TextMine'25, les modèles ont été entraînés sur la totalité du jeu de données d'entraînement afin de maximiser l'utilisation des données disponibles. Cette approche visait à renforcer la généralisation des modèles en exploitant pleinement les données additionnelles. Le jeu de données de test était divisé en deux parties : une partie publique, correspondant à un tiers du jeu de test, nous a permis d'évaluer nos solutions tout au long du défi tandis que la partie privée, fondée sur les deux tiers restants, est restée cachée jusqu'à la fin de la compétition. L'évaluation finale a été réalisée sur la macro-F1 uniquement. Pour les modèles évalués sans méthodes d'ensemble (premières lignes du tableau 8), il n'était pas possible de réaliser une évaluation préalable pour sélectionner le meilleur modèle étant donné

que ces modèles étaient entraînés sur la totalité du jeu d'entraînement. Ainsi, un modèle a été sélectionné aléatoirement parmi dix instances entraînées.

	Modèles et combinaisons de modèles testés	Score public	Score privé
	$\text{LACE}_{\text{mono}}$	62,7	61,2
	$\text{LACE}_{\text{multi}}$	**71,5**	**67,1**
	EIDER	34,8	32,2
	DREEAM	32,9	29,7
	Ensemble $\text{LACE}_{\text{mono}}$	66,3	62,0
(2)	Ensemble $\text{LACE}_{\text{multi}}$	71,3	66,3
	Ensemble EIDER	35,7	31,3
	Ensemble DREEAM	34,0	31,2
	Combinaison d'ensembles $\text{LACE}_{\text{multi}}$ + $\text{LACE}_{\text{mono}}$	67,8	63,1
	Combinaison d'ensembles $\text{LACE}_{\text{multi}}$ + EIDER	56,1	54,7
	Combinaison d'ensembles $\text{LACE}_{\text{multi}}$ + DREEAM	58,8	54,5
	Combinaison d'ensembles EIDER + DREEAM	35,7	31,6
(1)	Combinaison d'ensembles $\text{LACE}_{\text{multi}}$ + $\text{LACE}_{\text{mono}}$ + EIDER	70,6	65,4
	Combinaison d'ensembles $\text{LACE}_{\text{multi}}$ + $\text{LACE}_{\text{mono}}$ + DREEAM	70,5	65,3

TAB. 8 – *Scores macro-F1 des différents modèles, ensembles et combinaisons d'ensembles sur le jeu de données de test. (1) et (2) indiquent les soumissions sélectionnées respectivement comme premier et second choix pour le défi TextMine'25.*

Le tableau 8 montre les résultats de l'ensemble de nos soumissions. Le score sur la partie publique est toujours supérieur au score sur la partie privée, ce qui suggère une différence de distribution entre ces deux ensembles de données. Parmi les modèles soumis, $\text{LACE}_{\text{multi}}$ et son ensemble ont obtenu les meilleurs scores. Il est cependant important de noter que les performances de LACE peuvent légèrement varier d'une instance de modèle à l'autre (cf. tableau 6). En revanche, les modèles EIDER et DREEAM ont montré des performances nettement plus faibles que celles obtenues sur le jeu d'entraînement. Nous avons observé précédemment que ces modèles sont plus sensibles que LACE à la distribution des données d'entraînement, favorisant les classes majoritaires. Le résultat observé renforce donc l'idée que les données d'entraînement et les données de test ont des distributions différentes. Une analyse approfondie de la distribution des types de relations entre ces deux jeux de données pourrait apporter un éclairage supplémentaire sur ces résultats. Enfin, il faut noter que les soumissions que nous avons sélectionnées pour le défi TextMine'25 ne correspondent pas à nos meilleurs résultats. Nous avons en particulier privilégié des modèles d'ensemble en raison de la plus grande stabilité connue de ce type d'approches. Malgré la différence de distribution entre données publiques et privées que nous supposons au vu de certains de nos résultats, cette stratégie ne s'est pas avérée pertinente.

7 Conclusion

Cette étude a exploré plusieurs approches pour l'extraction de relations à l'échelle du document dans le cadre du défi TextMine'25. Nous avons expérimenté trois modèles fondés sur ATLOP : LACE, EIDER et DREEAM. Pour LACE, la façon dont sont construites les

représentations de couples d'entités et son utilisation en mono ou multi-étiquette ont été étudiées. Pour EIDER et DREEAM, des expériences sur les différentes manières de construire les preuves ont permis de montrer que la simple mise en place de règles heuristiques permet d'égaler une méthode plus lourde d'entraînement supervisé distant.

Le modèle LACE, grâce sa capacité à gérer les relations multi-étiquettes et à exploiter les interdépendances entre les relations grâce à un module de corrélation, a montré des performances supérieures dans un contexte few-shot. Cependant, les résultats en micro-F1 ont révélé que d'autres modèles, comme EIDER et DREEAM, sont plus efficaces pour les relations très représentées. D'autres expériences et une étude approfondie du jeu de test sont cependant nécessaires pour conclure sur la capacité de généralisation de ces deux modèles.

Afin de combiner les forces de ces différents modèles, nous avons d'abord tenté d'ajouter le module de corrélation des relations issu de LACE à DREEAM et EIDER. Ces expériences n'ayant pas donné de résultats positifs, nous avons utilisé une stratégie de méthodes d'ensemble par vote majoritaire. Bien que la combinaison de différents modèles reste une méthode énergivore et n'ait pas conduit à des améliorations significatives dans tous les cas, elle apporte une plus grande stabilité dans les résultats. Des recherches supplémentaires sur les techniques d'ensemble appliquées à l'extraction de relations pourraient offrir de nouvelles perspectives d'amélioration.

Références

Prieur, M., G. Gadek, H. M. Rawsthorne, A. Guille, P. Cuxac, et C. Lopez (2025). Défi TextMine'25 - Extraction de relations pour analyser des rapports de renseignement. In *Atelier TextMine'25 @ Extraction et Gestion des Connaissances 2025 (EGC'25)*.

Huang, Y. et Z. Lin (2024). Document-Level Relation Extraction with Relation Correlation Enhancement. In B. Luo, L. Cheng, Z.-G. Wu, H. Li, et C. Li (Eds.), *Neural Information Processing*, Singapore, pp. 427–440. Springer Nature, doi:10.1007/978-981-99-8178-6_33.

Zheng, Y., Y. Guo, Z. Luo, Z. Yu, K. Wang, H. Zhang, et H. Zhao (2023). A survey on document-level relation extraction : Methods and applications. In *Proceedings of the 3rd International Conference on Internet, Education and Information Technology (IEIT 2023)*, pp. 1061–1071. Atlantis Press, doi:10.2991/978-94-6463-230-9_128.

Ma, Y., A. Wang, et N. Okazaki (2023). DREEAM : Guiding attention with evidence for improving document-level relation extraction. In A. Vlachos et I. Augenstein (Eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Dubrovnik, Croatia, pp. 1971–1983. Association for Computational Linguistics, doi:10.18653/v1/2023.eacl-main.145.

Delaunay, J., H. T. H. Tran, C.-E. González-Gallardo, G. Bordea, N. Sidere, et A. Doucet (2023). A Comprehensive Survey of Document-level Relation Extraction (2016-2023). *arXiv preprint arXiv :2309.16396*, doi:10.48550/arXiv.2309.16396.

Xie, Y., J. Shen, S. Li, Y. Mao, et J. Han (2022). Eider : Empowering document-level relation extraction with efficient evidence extraction and inference-stage fusion. In *Findings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Tuo, A., R. Besançon, O. Ferret, et J. Tourille (2022). Better Exploiting BERT for Few-shot Event Detection. In P. Rosso, V. Basile, R. Martínez, E. Métais, et F. Meziane (Eds.), *27th International Conference on Applications of Natural Language to Information Systems (NLDB 2022)*, Valencia, Spain, pp. 291–298. Springer International Publishing.

Zhou, W., K. Huang, T. Ma, et J. Huang (2021). Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI conference on artificial intelligence*, Volume 35, pp. 14612–14620. doi:10.1609/aaai.v35i16.17717.

Yao, Y., D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, et M. Sun (2019). DocRED : A Large-Scale Document-Level Relation Extraction Dataset. In *57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 764–777. doi:10.18653/v1/P19-1074.

Christopoulou, F., M. Miwa, et S. Ananiadou (2019). Connecting the dots : Document-level neural relation extraction with edge-oriented graphs. In K. Inui, J. Jiang, V. Ng, et X. Wan (Eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, pp. 4925–4936. Association for Computational Linguistics, doi:10.18653/v1/D19-1498.

Veličković, P., G. Cucurull, A. Casanova, A. Romero, P. Liò, et Y. Bengio (2018). Graph attention networks. In *International Conference on Learning Representations*.

Smyth, P. et D. Wolpert (1997). Stacked density estimation. In M. Jordan, M. Kearns, et S. Solla (Eds.), *Advances in Neural Information Processing Systems*, Volume 10, pp. 668–674. MIT Press.

Freund, Y. et R. E. Schapire (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, San Francisco, CA, USA, pp. 148—-156. Morgan Kaufmann Publishers Inc.

Summary

In this article, we present a contribution from CEA-List to the TextMine'25 challenge, focusing on two main extensions of the ATLOP model, one taking into account correlations between relations, the other exploiting the notion of proof. We combine them through several ensemble configurations and compare the resulting combinations with these two extensions. The code corresponding to these experiments is available at the following address: `https://github.com/CEA-LIST/textmine-2025`.

Tackling Class Imbalance in Relation Extraction for french text: Effective Negative Sampling and Advanced Loss Functions

Iliass AYAOU*

*24 Bd de la Victoire, 67000 Strasbourg
iliass.ayaou@insa-strasbourg.fr
https://orcid.org/0009-0007-5247-8347

Abstract. Relation extraction is crucial for understanding semantic relationships in text, yet highly imbalanced datasets pose significant challenges, especially when minority classes have very few instances. In the TEXTMINE 2025 challenge, we addressed these issues by formulating the task as a multi-label, multiclass text classification problem. Our approach combines effective negative sampling with a ratio of 3:1, adaptive class weighting using the Effective Number of Samples method, advanced loss functions such as focal loss and asymmetric loss, and per-class threshold optimization. We also introduced special tokens to mark entities and relations, which helped avoid text truncation due to token limits and improved the model's ability to capture self-reflective relations like `HAS_GENDER_MALE` and `HAS_GENDER_FEMALE`. Experiments conducted on a single Tesla V100 GPU with 32 GB of VRAM demonstrated that DistilCamemBERT offered the best efficiency, while CamemBERT base achieved the highest macro F1 score. Our methods significantly improved performance on underrepresented relations, demonstrating the effectiveness of our approach in handling highly imbalanced datasets.

1 Introduction

Relation extraction (RE) is a fundamental task in natural language processing (NLP) that involves identifying semantic relationships between entities within text. It plays a crucial role in various applications, such as information retrieval, question answering, and knowledge base construction. However, RE models often face challenges when dealing with highly imbalanced datasets, where some relation types are significantly underrepresented.

In the TEXTMINE 2025 challenge, participants were tasked with developing methods to extract relations from a French text corpus. The dataset presented a severe class imbalance problem, with some relations like `WAS_DISSOLVED_IN` having only 15 occurrences, while others like `IS_LOCATED_IN` had thousands of instances. This imbalance poses a significant challenge for models, as they tend to be biased toward majority classes, leading to poor performance on minority classes.

To address this problem, we formulate the relation extraction task as a multi-label, multiclass text classification problem. This formulation allows each entity pair to have multiple possible relations, which is suitable for capturing complex relationships in the dataset.

Our approach combines several techniques to address class imbalance and improve performance:

- **Effective Negative Sampling**: We experimented with multiple negative-to-positive sampling ratios and found that a 3:1 ratio yielded the best results. By generating hard negative samples (entity pairs that could have a relation but do not) and introducing a `no_relation` class (not present in the original dataset), we taught the model to distinguish between entity pairs that have a relation and those that do not.
- **Advanced Loss Functions**: We utilized focal loss and asymmetric loss to focus the model's learning on underrepresented classes. The asymmetric loss with gamma parameters ($\gamma_{\text{neg}} = 4$, $\gamma_{\text{pos}} = 1$) yielded slightly better results than focal loss, improving the macro F1 score by approximately 0.005.
- **Per-Class Threshold Optimization**: We optimized decision thresholds for each class individually, enhancing the balance between precision and recall, especially for minority classes.
- **Special Tokens and Entity Marking**: We introduced special tokens to mark entities and relations within the text. This not only helped the model to identify the roles of entities but also avoided text truncation due to the 512-token limit of the models, as the special tokens are not split by the tokenizer. Additionally, we introduced special markers for self-reflective relations like `HAS_GENDER_MALE` and `HAS_GENDER_FEMALE`.
- **Model Selection and Training**: We tested several models, including DistilCamemBERT, CamemBERT base, Flaubert base, and XLM-RoBERTa. DistilCamemBERT was the most efficient in terms of size and training time, while CamemBERT base achieved the best results. Experiments were conducted on a single Tesla V100 GPU with 32 GB of VRAM.

Our approach follows the common paradigm for adapting large pretrained language models (PLMs)—such as CamemBERT or DistilCamemBERT—to specific downstream tasks: we attach a classification head on top of the transformer layers and backpropagate using a task-specific loss.

Our experiments showed that the proposed approach significantly improves performance on underrepresented relations, demonstrating its effectiveness in handling highly imbalanced datasets.

2 Related Work

Relation Extraction and Class Imbalance. Relation extraction has been widely studied in NLP, with early approaches relying on feature engineering and statistical methods Culotta and Sorensen (2004). The introduction of deep learning led to models that automatically learn features from data, such as CNNs and RNNs Zeng et al. (2014).

Class imbalance is a common issue in machine learning, particularly in tasks like RE where certain relations are rare. Traditional methods to address class imbalance include resampling techniques Chawla et al. (2002), cost-sensitive learning Elkan (2001), and data augmentation Wei and Zou (2019).

Negative Sampling. Negative sampling is a technique used to provide models with negative examples during training Mikolov et al. (2013). In RE, negative sampling helps the model learn to distinguish between actual relations and coincidental co-occurrences of entities. Hard negative mining, where challenging negative examples are selected, has been shown to improve model robustness Schroff et al. (2015).

Advanced Loss Functions. Loss functions like focal loss Lin et al. (2017) and asymmetric loss Ridnik et al. (2021) have been proposed to address class imbalance by focusing the training on hard-to-classify examples or underrepresented classes. These loss functions adjust the contribution of each sample to the loss based on the difficulty of classification.

Use of Special Tokens. Special tokens have been used to provide models with additional context or to mark specific parts of the input Lin and Ji (2014); Soares et al. (2019). In RE, marking entities within the text can help the model understand their roles and relationships.

Our work builds upon these foundations by combining effective negative sampling, advanced loss functions, per-class threshold optimization, and the use of special tokens to improve performance on highly imbalanced RE tasks.

3 Methodology

Task Formulation. We formulated the relation extraction task as a *multilabel*, *multiclass* text classification problem. In this setup, each entity pair within a text can have multiple relations assigned to it. This formulation is suitable for capturing the complexity of relationships in the dataset and allows the model to predict multiple relations for a given pair.

3.1 Data Preprocessing

Data Loading and Parsing. We loaded the dataset, which consisted of texts with annotated entities and relations. Each text included a list of entities, each with unique identifiers, types, and mentions (including start and end positions within the text). Relations were provided as triplets: (*head entity ID*, *relation*, *tail entity ID*).

Entity Extraction and Preprocessing. For each entity, we collected all its mentions and sorted them based on their starting positions. We then normalized the text by converting it to lowercase to ensure consistency.

Unique Relation Mapping. We extracted all unique relations present in the training data by iterating through each text's relations and capturing the combinations of head entity type, relation type, and tail entity type. This process highlighted the severe class imbalance, as some relations had very few occurrences.

3.2 Negative Sampling Strategy.

Negative-to-Positive Ratio. We experimented with negative-to-positive sampling ratios of 1:1, 2:1, and 3:1. The ratio of 3:1 provided the best macro F1 score, balancing the need for hard negatives and training efficiency.

Hard Negative Generation. We generated hard negatives by pairing entities that could potentially have a relation according to the mappings we extracted but did not in the annotations. For instance, if two entities could have a `LOCATED_IN` relation based on their types but were not annotated as such, we included them as negative examples.

Introduction of `no_relation` Class. We introduced a `no_relation` class (not present in the original training dataset) to explicitly teach the model about the absence of relations between entity pairs. This helped the model learn to distinguish between entity pairs that have a relation and those that do not.

3.3 Special Tokens and Entity Marking

Avoiding Text Truncation. The models we used have a 512-token limit. To avoid truncating important parts of the text, we introduce special tokens to mark entities and relations. These special tokens are not split by the tokenizer, which helps in efficiently utilizing the model's context window.

Entity and Relation Markers. We mark entities using special tokens that include the entity's role and type. For instance, we used `[e1:ENTITY_TYPE]` and `[e2:ENTITY_TYPE]` to mark the head and tail entities, respectively, and the closing tokens `[/e1:ENTITY_TYPE]` and `[/e2:ENTITY_TYPE]`.

We also append possible relation hints at the end of the text using special tokens like `[rel:RELATION_TYPE]`. This provided the model with additional context on potential relations between the entities.

Self-Reflective Relations. For self-reflective relations where the head and tail entities are the same, such as `HAS_GENDER_MALE` and `HAS_GENDER_FEMALE`, we introduced special markers like `[se:ENTITY_TYPE]` and `[/se:ENTITY_TYPE]`. This helped the model handle these special cases more effectively.

3.4 Handling Class Imbalance

Adaptive Class Weighting. We computed class weights using the Effective Number of Samples method Cui et al. (2019), which assigns higher weights to minority classes based on their relative frequencies. The class weight for each class was calculated using:

$$w_i = \frac{1 - \beta}{1 - \beta^{n_i}}$$

where n_i is the number of samples for class i, and β is a hyperparameter set to 0.9999.

We adjusted the weight for the `no_relation` class by setting it to the minimum weight among all classes. This prevented the dominant `no_relation` class from overpowering the loss function and ensured that minority classes received adequate attention during training.

Advanced Loss Functions. We experiment with the following loss functions:

- **Focal Loss**:

  Focal loss Lin et al. (2017) is designed to address class imbalance by focusing training on hard-to-classify examples. The focal loss function is defined as:

  $$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)$$

  where p_t is the model's estimated probability for the true class, α_t is the class weight, and γ is the focusing parameter.

- **Asymmetric Loss**:

  We also experimented with the asymmetric loss function Ridnik et al. (2021), which is a variant of the focal loss designed specifically for imbalanced multilabel classification. The asymmetric loss introduces separate focusing parameters for positive and negative samples. We set the parameters to $\gamma_{\text{neg}} = 4$ and $\gamma_{\text{pos}} = 1$, which yielded slightly better results than the focal loss, improving the macro F1 score by approximately 0.005.

3.5 Per-Class Threshold Optimization

We implemented per-class threshold optimization to improve the balance between precision and recall for each class. This process involved:

- **Collect validation logits**: After applying $\sigma(\cdot)$ for each class, we sample thresholds in $\{0.1, 0.15, \ldots, 0.9\}$.
- **Compute F1 per threshold per class**: The threshold maximizing F1 on validation data is stored.
- **Applying Optimal Thresholds**: During inference, we applied these per-class thresholds to the model's output probabilities to determine the final predictions.

This approach allowed us to fine-tune the decision boundary for each class individually, which was particularly beneficial for minority classes where the default threshold (e.g., 0.5) might not be optimal.

3.6 Model Architecture and Training

Model Selection. We tested several pretrained transformer models to find the best balance between performance and efficiency:

- **DistilCamemBERT** Martin et al. (2020): A distilled version of CamemBERT that is smaller and faster while retaining good performance. It was the most efficient model in terms of size and training time required.
- **CamemBERT Base** Martin et al. (2020): Achieved the best macro F1 score among the tested models.
- **Flaubert Base** Le et al. (2019): A French language model based on the RoBERTa architecture.
- **XLM-RoBERTa** Conneau et al. (2020): A multilingual model capable of handling multiple languages, including French.

Training Environment. All experiments were conducted on a single Tesla V100 GPU with 32 GB of VRAM. This setup provided sufficient computational resources to train the models efficiently.

Hyperparameters and Regularization. We performed hyperparameter tuning to optimize model performance:

- **Learning Rate**: Set to 2×10^{-5}.
- **Batch Size**: 16 for both training and evaluation.
- **Number of Epochs**: Trained for up to 10 epochs with early stopping based on validation performance.
- **Weight Decay**: Set to 0.01. We observed that lowering the weight decay led to overfitting.
- **Dropout Rates**: Adjusted dropout rates to prevent overfitting. Removing dropout or lowering it resulted in overfitting, so appropriate dropout was necessary.

We used the AdamW optimizer Loshchilov and Hutter (2019) with weight decay and enabled mixed precision training (fp16) to accelerate computations.

4 Experiments and Results

4.1 Dataset Description

The dataset provided in the TEXTMINE 2025 challenge consisted of texts in French with annotated entities and relations. The class distribution was highly imbalanced:

- **Majority Classes**: Relations like `IS_LOCATED_IN` had thousands of occurrences.

- **Minority Classes**: Relations like `WAS_DISSOLVED_IN` had only 15 occurrences.

This imbalance presented a significant challenge for training models that perform well across all classes.

4.2 Evaluation Metrics

We used the **macro F1 score** as the primary evaluation metric, as it gives equal weight to all classes, including minority ones.

4.3 Results

Negative Sampling Ratio. We experimented with negative-to-positive ratios of 1:1, 2:1 and 3:1. The ratio of 3:1 provided the best macro F1 score, balancing the need for hard negatives and training efficiency. If no negatives are presented, the precision of the model drops significantly.

TAB. 1 – *Comparison of Loss Functions*

Loss Function	**Macro F1 Score**	**Improvement**
Cross-Entropy	0.662	—
Focal Loss	0.682	+0.020
Asymmetric Loss	**0.687**	**+0.025**

Loss Function Comparison. Using asymmetric loss with $\gamma_{\text{neg}} = 4$ and $\gamma_{\text{pos}} = 1$ yielded the best performance, slightly improving over focal loss.

TAB. 2 – *Model Performance Comparison*

Model	**Macro F1 Score**	**Training Time**
DistilCamemBERT	0.660	1h 14min
CamemBERT Base	**0.687**	2h 24min
Flaubert Base	0.662	3h 09min
XLM-RoBERTa	0.650	2h 44min

Model Performance (using asymmetric loss). CamemBERT base achieved the highest macro F1 score, but required more training time than DistilCamemBERT. DistilCamemBERT provided the best efficiency.

Impact of Special Tokens. Incorporating special tokens to mark entities and relations resulted in an increase of approximately 0.02 in macro F1 score compared to models without entity marking. The special tokens helped the model to better understand the context and relationships between entities.

Per-Class Threshold Optimization Results. Per-class threshold optimization improved the macro F1 score by adjusting the decision thresholds for each class individually. For example, the optimal threshold for the minority class `WAS_DISSOLVED_IN` was lower than the default 0.5, allowing the model to make more positive predictions for this underrepresented class without significantly increasing false positives.

5 Discussion

Effectiveness of Negative Sampling. The hard negative samples generated by our negative sampling strategy helped the model learn to distinguish between entity pairs that have a relation and those that do not. Introducing the `no_relation` class was crucial in this aspect.

Impact of Advanced Loss Functions. The use of focal loss and asymmetric loss focused the model's learning on underrepresented classes by adjusting the loss contribution of hard-to-classify examples. Asymmetric loss provided a slight edge over focal loss.

Role of Special Tokens. The special tokens not only allowed for explicit entity and relation marking but also prevented important information from being truncated due to token limits. This enhanced the model's ability to capture long-range dependencies and contextual information.

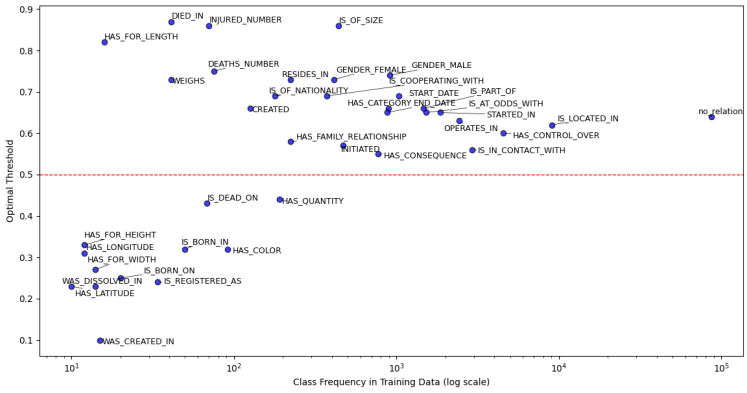


FIG. 1 – *Class Frequency and obtained optimal thresholds*

Per-Class Threshold Optimization. Per-class threshold optimization proved essential in fine-tuning the model's performance for each class. By adjusting the thresholds based on validation data, we balanced precision and recall according to the characteristics of each class, which was particularly beneficial for minority classes. We can also see that for some classes (such as `DIED_IN`), the optimal threshold value found is quite high despite the relatively low

frequency of the class in the dataset. This suggests that the data format as well as the variability of the context can affect how hard it is to over- or under-predict, further motivating the choice of this strategy over a simple frequency-based threshold optimization.

Model Selection and Efficiency. DistilCamemBERT was efficient in terms of training time and resource usage, making it suitable for scenarios with limited computational resources. CamemBERT base, while requiring more resources, achieved the best performance.

Regularization and Overfitting. We observed that removing dropout or lowering weight decay led to overfitting, particularly on the majority classes. Appropriate regularization was necessary to ensure the model generalized well to unseen data.

6 Conclusion

We presented an approach to address class imbalance in relation extraction tasks by combining effective negative sampling, advanced loss functions, per-class threshold optimization, and the use of special tokens. By formulating the task as a multilabel, multiclass text classification problem and introducing techniques to focus on minority classes, we significantly improved the macro F1 score, particularly on underrepresented relations.

Our experiments demonstrated that DistilCamemBERT provided the best efficiency, while CamemBERT base achieved the highest performance. The methods we employed are effective for handling highly imbalanced datasets and can be applied to other NLP tasks facing similar challenges.

Some possible improvements can be introduced by creating a more detailed relation mapping to avoid generating too many hard negatives as well as introducing advanced data augmentation techniques that can generate realistic entries for minority classes. Another possible direction to explore is to combine two models where the first one will be tasked to predict whether two entities have a relation or not while the second one would handle predicting the relations.

Acknowledgments

We would like to thank the organizers of the TEXTMINE 2025 challenge for providing the dataset and the opportunity to work on this interesting problem.

References

Ridnik, T., E. Ben-Baruch, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor (2021). Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 82–91.

Conneau, A., K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 8440–8451.

Martin, L., B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de la Clergerie, D. Seddah, and B. Sagot (2020). Camembert: A tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7203–7219.

Soares, L. B., N. FitzGerald, J. Ling, and T. Kwiatkowski (2019). Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2895–2905.

Le, H., E. Santus, and A. Moschitti (2019). Flaubert: Unsupervised contextualized embeddings for french. *arXiv preprint arXiv:1909.02121*.

Wei, J. and K. Zou (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6382–6388.

Loshchilov, I. and F. Hutter (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Cui, Y., M. Jia, T.-Y. Lin, Y. Song, and S. Belongie (2019). Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9268–9277.

Lin, T.-Y., P. Goyal, R. Girshick, K. He, and P. Dollár (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988.

Schroff, F., D. Kalenichenko, and J. Philbin (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823.

Zeng, D., K. Liu, S. Lai, G. Zhou, and J. Zhao (2014). Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014*, pp. 2335–2344.

Lin, Y. and H. Ji (2014). Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 402–412.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Culotta, A. and J. Sorensen (2004). Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 423–429.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16*, 321–357.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 973–978.

Résumé

Relation extraction is crucial for understanding semantic relationships in text, yet highly imbalanced datasets pose significant challenges, especially when minority classes have very few instances. In the TEXTMINE 2025 challenge, we addressed these issues by formulating

the task as a multilabel, multiclass text classification problem. Our approach combines effective negative sampling with a ratio of 3:1, adaptive class weighting using the Effective Number of Samples method, advanced loss functions such as focal loss and asymmetric loss, and per-class threshold optimization. We also introduced special tokens to mark entities and relations, which helped avoid text truncation due to token limits and improved the model's ability to capture self-reflective relations like `HAS_GENDER_MALE` and `HAS_GENDER_FEMALE`. Experiments conducted on a single Tesla V100 GPU with 32 GB of VRAM demonstrated that DistilCamemBERT offered the best efficiency, while CamemBERT base achieved the highest macro F1 score. Our methods significantly improved performance on underrepresented relations, demonstrating the effectiveness of our approach in handling highly imbalanced datasets.

GLiDRE : Modèle généraliste pour l'extraction de relations à l'échelle de documents

Robin Armingaud*

*Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
{nom.prenom}@cea.fr,
https://kalisteo.cea.fr/

Résumé. Cet article décrit notre contribution au défi TextMine'25, qui consiste à adapter le modèle GLiNER (Zaratiana et al., 2023) à la tâche d'extraction de relations. Notre méthode s'appuie sur deux encodeurs : gte-multilingual-base pour les labels et XLM-RoBERTa pour les documents, complétés par une représentation locale inspirée de l'approche ATLOP (Zhou et al., 2021). Une phase de pré-entraînement sur des données synthétiques a permis d'optimiser les performances. Notre meilleure soumission atteint un macro F1 score de 0,578 sur le jeu de test privé. Bien que ce score soit inférieur à celui des meilleures contributions, il illustre le potentiel de GLiDRE dans des contextes peu dotés.

1 Introduction

L'extraction de relations constitue une tâche clé en traitement automatique du langage naturel, mais elle reste un défi complexe, en particulier à l'échelle du document. Peu de jeux de données abordent cette tâche, le plus utilisé étant DocRED (Yao et al., 2019), disponible uniquement en langue anglaise. Le défi TextMine 2025 (Prieur et al., 2025) cible ainsi l'extraction de relation dans des documents de renseignement, domaine pour lequel l'identification des interactions complexes entre les acteurs est cruciale et est encore aujourd'hui majoritairement effectuée manuellement. Le corpus proposé comprend 800 rapports de renseignement factices annotés selon 37 types de relations spécifiques. Il est en langue française, qui est beaucoup moins dotée en termes de modèles de fondation et d'ensemble de données. Motivés par les performances de l'architecture GLiNER (Zaratiana et al., 2023), en particulier lors du challenge Evalllm 2024 (Armingaud et al., 2024), nous avons développé GLiDRE (Generalist and Lightweight Model for Document-level Relation Extraction). Cette version bi-encodeur est adaptée pour répondre aux exigences spécifiques de l'extraction de relations à l'échelle du document. GLiDRE intègre une représentation locale des entités basée sur les mécanismes d'attention inspirés d'ATLOP (Zhou et al., 2021) et bénéficie d'une phase de pré-entraînement sur un ensemble de données synthétiques, optimisant ainsi son efficacité dans des contextes peu dotés.

2 Méthode proposée

2.1 Modèle

GLiNER GLiNER est un modèle de reconnaissance d'entités nommées basé sur un encodeur de type BERT (Devlin, 2018). Il associe les plongements lexicaux des types d'entité aux mentions d'entité correspondantes, offrant une grande flexibilité pour traiter divers types d'entités. GLiNER a originellement été entraîné sur un ensemble de données varié et synthétique généré par ChatGPT, et se distingue par ses performances en *zero-shot* (sans exemple spécifique de la tâche durant l'entraînement) ou *few-shot* (avec un très faible nombre d'exemples de la tâche durant l'entraînement), motivant son adaptation pour les types de relation de TextMine 2025 comportant un faible support.

GLiDRE Bien que plusieurs adaptations de GLiNER à l'extraction de relations aient été proposées [1] [2], aucune ne cible spécifiquement l'échelle du document ni n'exploite les coréférences fournies dans le cadre du challenge. Nous proposons une adaptation de GLiNER en nous basant sur sa version bi-encodeur [3], qui sépare l'encodeur du texte et celui des étiquettes pour étendre la fenêtre de contexte. L'encodeur des étiquettes s'appuie sur le modèle *gte-multilingual-base* (Zhang et al., 2024), tandis que l'encodeur du texte utilise *xlm-roberta-large* (Conneau, 2019)(Tab. 1).

Encodeur labels	Encodeur texte	Public	Privé	Moyenne
gte-multilingual-base (multi)	xlm-roberta- large (multi)	**0.602**	**0.543**	**0.5725**
bge-large-en-v1.5 (en)	roberta-large (en)	0.552	0.526	0.539
sentence-camembert-large (fr)	camembert-large (fr)	0.593	0.526	0.5595
bge-large-en-v1.5 (en)	camembert-large (fr)	0.591	0.518	0.5545

TAB. 1 – *Comparaison en macro-F1 sur les ensembles de test du défi de différents modèles de fondation. Les labels sont en anglais et les textes en français. Les modèles multilingues sont les plus performants. Les modèles sont de taille similaire pour une comparaison équitable.*

Notre modèle compare les représentations des relations avec celles des types de relations associés (Fig. 1).

Une relation est définie comme une paire d'entités, chaque entité étant composée d'une liste de mentions correspondant à ses différentes références dans le texte. Nous calculons d'abord les représentations des mots en utilisant le plongement lexical du premier token constituant chaque mot, puis nous moyennons ces représentations pour obtenir des représentations des mentions. Nous moyennons à nouveau les représentations de chaque mention pour obtenir les représentations des entités. Enfin, la représentation de la relation est obtenue en concaténant la représentation des deux entités puis en la projetant à travers une couche linéaire pour obtenir une représentation de la même dimension que celle des entités (Fig. 2).

Comme fonction de coût, nous utilisons la *focal loss* (Lin, 2017), qui surpasse l'entropie croisée binaire (*BCE loss*) en performance en cas de déséquilibre de classe (Tab. 2).

1. https://github.com/jackboyla/GLiREL
2. https://huggingface.co/knowledgator/gliner-multitask-large-v0.5
3. https://huggingface.co/knowledgator/gliner-bi-large-v1.0

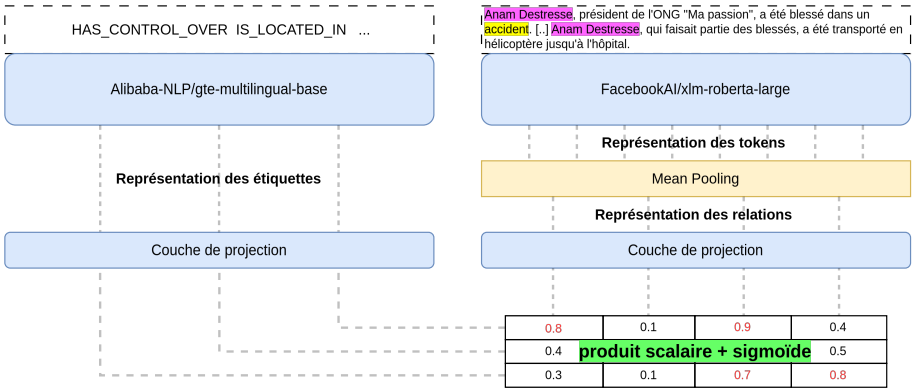


FIG. 1 – *Architecture de GLiDRE. Les représentations des labels et des relations sont projetées dans le même espace et les paires relation-étiquette dont la similarité excède un seuil sont prédites.*

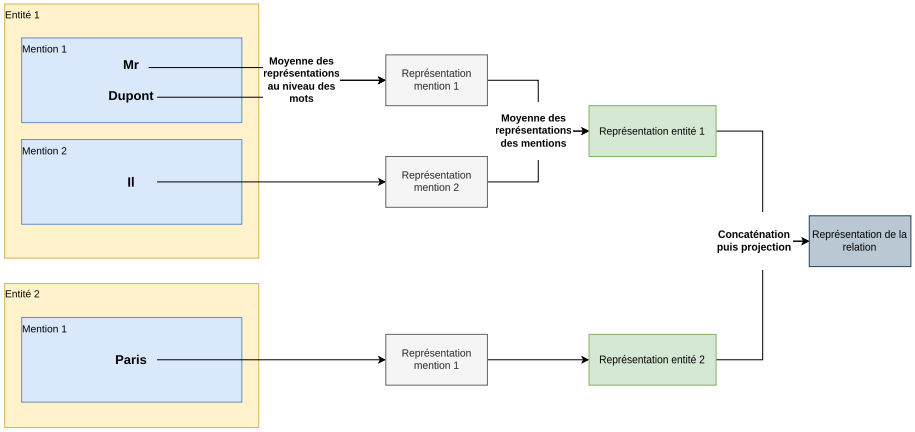


FIG. 2 – *Construction des représentations des relations. Les plongements lexicaux des mots, des mentions puis des entités constituant une relation sont combinés pour obtenir une représentation globale.*

Fonction de coût	Public	Privé	Moyenne
Focal Loss	**0.602**	**0.543**	**0.5725**
BCE Loss	0.510	0.494	0.502

TAB. 2 – *Comparaison de différentes fonctions de coût en macro-F1.*

2.2 Pré-entraînement

Afin de tenter d'améliorer les performances du modèles, nous avons constitué un corpus synthétique de pré-entraînement. Nous avons entraîné un classifieur *fasttext* (Bojanowski et al., 2017) avec comme classe positive les textes du défi et comme classe négative des textes aléatoires issus de la section française du large corpus issus d'internet OSCAR (Suárez et al., 2019). Nous avons ensuite filtré l'ensemble d'OSCAR pour ne conserver que les documents dont la longueur n'excède pas 512 tokens et dont le score de confiance du classifieur est supérieur à 0,9. Le corpus de textes similaires à ceux du challenge ainsi créé regroupe environ 30 000 documents. Ces documents sont ensuite annotés automatiquement avec des entités du challenge en deux étapes. La première étape consiste à *prompter* (donner des instructions en langage naturel à un large modèle de langage) *Llama-3.1 8B Instruct* (Dubey et al., 2024) afin qu'il annote automatiquement les entités et leurs coréférences dans le texte en fournissant un exemple issu de l'ensemble d'entraînement. La deuxième étape consiste à prompter ce même modèle pour qu'il annote les entités identifiées avec les relations du défi. Nous filtrons ensuite les annotations incorrectes (hallucination du type de relation ou format de données non valide) pour obtenir finalement un ensemble d'environ 25000 documents annotés automatiquement. Nous pré-entrainons notre modèle sur 400 000 étapes.

2.3 Contexte local

Les méthodes à l'état de l'art (Xie et al., 2021; Ma et al., 2023) intègrent une représentation locale introduite par Zhou et al. (2021) dans l'approche ATLOP. Nous l'avons intégré à notre modèle de manière similaire en concaténant aux représentations des entités une représentation globale du document pondérée par les attentions du modèle.

2.4 Hyperparamètres

Les hyperparamètres utilisés pour les entraînements sur les documents du défi sont les suivants :

Paramètre	**Valeur**
Encodeur texte	xlm-roberta-large
Encodeur labels	gte-multilingual-base
Hidden size	512
Dropout	0.4
Nombre d'étapes	40000
Batch size	1
Gradient accumulation steps	8
Warmup ratio	0.1
Scheduler	Cosine
Pooling	Mean
Alpha focal loss	0.8
Gamma focal loss	3
Learning rate encodeur	1e-5
Learning rate autre	1e-4
Weight decay encodeur	0.01
Weight decay autre	0.01

3 Résultats

Nous appliquons un filtre sur les types de relations possibles, conformément aux informations fournies avec le jeu de données POPCORN[4]. Le seuil de prédiction est optimisé pour chaque relation sur un ensemble de développement correspondant à une séparation 80/10/10 de l'ensemble d'entraînement.

Méthode	Public	Privé	Moyenne
GLiDRE pretrain (zero-shot, late submission)	0.319	0.296	0.3075
GLiDRE (base)	**0.602**	0.543	0.5725
+ Représentation locale (late submission)	0.545	0.53	0.5375
+ Pré-entraînement (late submission)	0.561	0.564	0.5625
+ Représentation locale + Pré-entraînement	0.600	**0.578**	**0.589**

TAB. 3 – *Macro-F1 sur l'ensemble de test du challenge.*

Ce modèle obtient des performances honorables, ce qui indique que l'approche est prometteuse (même si les scores restent en dessous des meilleurs modèles de la compétition), en particulier si nous étions dans un contexte *zero-shot* ou *few-shot*. L'impact du pré-entraînement seul sur les performances moyennes reste limité. Cette observation peut s'expliquer par la taille suffisante du jeu d'entraînement fourni dans le cadre du défi, ce qui réduit l'intérêt d'un pré-entraînement sur des données synthétiques. Les auteurs de GLiNER observent de manière similaire une baisse de l'efficacité du pré-entraînement quand le jeu de données d'entraînement augmente. La représentation locale utilisée seule diminue les performances, mais ces résultats doivent être interprétés avec précaution. En effet, des erreurs dans l'implémentation d'ATLOP

4. https://github.com/Emvista/popcorn-dataset

n'ont pas pu être corrigées avant la fin de la compétition, ce qui pourrait avoir biaisé les résultats. Notre meilleur modèle combine les deux améliorations proposées et atteint un score moyen de 0.589 entre le jeu de données de test public et privé, soit la dixième position sur les quinze participants au défi inscrit sur la plateforme Kaggle. Ces résultats encouragent de futurs travaux évaluant plus précisément l'impact du contexte local et une évaluation plus détaillée de GLiDRE dans un contexte *zero-shot* ou *few-shot*.

Références

Prieur, M., G. Gadek, A. Guille, H. Rawsthorne, P. Cuxac, et C. Lopez (2025). Défi textmine'25 - extraction de relations pour analyser des rapports de renseignement. *Actes de l'atelier TextMine'25, p. à paraître, Extraction et Gestion des Connaissances 2025 (EGC'25)*.

Zhang, X., Y. Zhang, D. Long, W. Xie, Z. Dai, J. Tang, H. Lin, B. Yang, P. Xie, F. Huang, et al. (2024). mgte : Generalized long-context text representation and reranking models for multilingual text retrieval. *arXiv preprint arXiv :2407.19669*.

Dubey, A., A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv :2407.21783*.

Armingaud, R., A. Peuvot, R. Besançon, O. Ferret, S. Souihi, et J. Tourille (2024). Cea-list@ evalllm2024 : prompter un très grand modèle de langue ou affiner un plus petit ? *Atelier sur l'évaluation des modèles génératifs (LLM) et challenge d'extraction d'information few-shot*.

Zaratiana, U., N. Tomeh, P. Holat, et T. Charnois (2023). Gliner : Generalist model for named entity recognition using bidirectional transformer. *arXiv preprint arXiv :2311.08526*.

Ma, Y., A. Wang, et N. Okazaki (2023). Dreeam : Guiding attention with evidence for improving document-level relation extraction. *arXiv preprint arXiv :2302.08675*.

Xie, Y., J. Shen, S. Li, Y. Mao, et J. Han (2021). Eider : Empowering document-level relation extraction with efficient evidence extraction and inference-stage fusion. *arXiv preprint arXiv :2106.08657*.

Zhou, W., K. Huang, T. Ma, et J. Huang (2021). Document-level relation extraction with adaptive thresholding and localized context pooling. *Proceedings of the AAAI conference on artificial intelligence 35*(16), 14612–14620.

Conneau, A. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv :1911.02116*.

Yao, Y., D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, et M. Sun (2019). Docred : A large-scale document-level relation extraction dataset. *arXiv preprint arXiv :1906.06127*.

Suárez, P. J. O., B. Sagot, et L. Romary (2019). Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*.

Devlin, J. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

Lin, T. (2017). Focal loss for dense object detection. *arXiv preprint arXiv :1708.02002*.

Bojanowski, P., E. Grave, A. Joulin, et T. Mikolov (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics 5*, 135–146.

Summary

This work adapts the GLiNER model for document-level relation extraction in French, as part of the TextMine 2025 challenge. Enhancements include pretraining on a subset of OS-

GLiDRE

CAR dataset, local representations inspired by ATLOP, and optimized prediction thresholds. Results demonstrate modest performance but the model demonstrates potential in low-resource scenarios.

Affinage de Transformers et Larges Modèles de Langage pour l'Extraction de Relations Synthétiques (TextMine 2025)

Jean Meunier-Pion*

*CentraleSupélec, 3 Rue Joliot-Curie, 91190 Gif-sur-Yvette
jean.meunier-pion@centralesupelec.fr,
https://www.centralesupelec.fr

Résumé. Pour le défi TextMine 2025 sur l'extraction de relations à partir de rapports synthétiques, nous proposons une approche hybride combinant des larges modèles de langage et l'affinage de transformers pour classifier des segments de texte en relations, atteignant un F1-score de 64,0% sur les données de test privées.

1 Introduction

Dans le cadre du défi TextMine 2025 (Prieur et al. (2025)), portant sur l'extraction de relations dans des rapports synthétiques, nous proposons une approche combinant deux méthodes : une méthode supervisée pour les classes de relations fréquentes et une méthode sans entraînement pour les classes rares. La méthode supervisée utilise une architecture transformer pré-entraînée et intègre un module prenant en compte l'information spécifique au sujet, à l'objet, et au contexte relationnel, ainsi que leurs interactions deux à deux. La méthode sans entraînement repose sur des larges modèles de langage pré-entraînés. Cette combinaison nous a permis d'atteindre un F1-score de 64.0% sur l'ensemble de test privé, en tenant compte du caractère multi-étiquette des données, des déséquilibres entre les classes, et du type des entités et relations. Le code sera rendu public à l'adresse : https://github.com/jmpion/textmine25

2 Problématique

Les rapports contiennent des entités annotées, et chaque paire (e_s, e_o) doit être associée à une ou plusieurs relations $\tilde{r} \in R$, où R comprend 37 classes de relations et une classe vide. L'aspect multi-étiquette est un enjeu non-négligeable : 18,6% des paires avec au moins une relation en ont au moins deux différentes. Ici, le choix est fait de concevoir un modèle de classification binaire par classe de relations, plutôt qu'un modèle global faisant de la classification multiple, afin de disposer de flexibilité dans la sélection et la combinaison des méthodes, et de gérer plus simplement les déséquilibres de données. L'objectif du défi est de maximiser le F1-score macro sur l'ensemble des classes de relations.

À noter, le choix de créer un modèle par classe de relation a été fait pour plusieurs raisons : (1) éviter le problème de déséquilibre des données inter-classes, en se focalisant sur seulement une classe à la fois (2) éviter le problème de classification multi-étiquette, en particulier pour

simplifier la gestion des déséquilibres de données, et (3) bénéficier de davantage de modularité, en développant un modèle personnalisé pour chaque type de relation.

3 Données

Le jeu de données mis à disposition pour ce challenge est le jeu POPCORN (**P**euplement **OP**érationnel de **CO**nnaissances et **R**éseaux **N**euronaux) introduit par Giordano et al. (2024). Pour ce défi, il était divisé en un ensemble d'entraînement de 800 exemples et un ensemble de 400 exemples de test. Sur les 400 exemples de test, 200 étaient alloués au classement public du défi Kaggle (Prieur et al. (2025)), tandis que 200 autres étaient utilisés pour le classement privé final. Chaque exemple est composé d'un identifiant unique, d'un texte, d'une liste d'entités, et d'une liste de relations. Les données disponibles dans chaque sont l'identifiant, le texte, et la liste d'entités. Les données à prédire sont les listes de relations, accessibles dans le jeu d'entraînement, mais cachées pour le jeu de test. L'objectif est de prédire, pour chaque exemple, la liste des relations, à partir de l'identifiant, du texte, et des entités.

3.1 Structure de la connaissance

Les classes d'entités et de relations sont listées dans des tables d'entités, d'attributs, et de relations, fournies sur la page *Overview* de la compétition Kaggle du défi (Prieur et al. (2025)) et sont également accessibles à partir du jeu d'entraînement. En tout, il y a 37 classes de relations. Il y a quatre types d'entités : ACTOR, EVENT, MATERIEL, et PLACE. Les types ACTOR et EVENT ont ensuite plusieurs sous-classes et niveaux de sous-classes. Il y a 10 attributs, dont deux, TIME et QUANTITY sont subdivisés en quatre sous-types distincts permettant de distinguer lorsque les valeurs sont des minimums, des maximums, exactes, ou floues.

Au niveau des relations, les 37 classes ont chacune un type contraignant le type de l'entité sujet et le type de l'entité objet de la relation. Par exemple, la relation IS_PART_OF ne peut avoir lieu qu'entre une entité sujet de type ACTOR et une entité objet de type ORGANIZATION, qui est un sous-type de ACTOR. Cependant, quelques types de relations fournis dans la table des relations susmentionnée n'étaient pas exacts par rapport aux données du jeu d'entraînement. C'est ainsi le cas de la classe de relation CREATED pour laquelle le type indiqué dans la table est (ACTOR, ORGANIZATION), mais on observe dans le jeu d'entraînement 100 relations de la forme (ACTOR, CREATED, MATERIEL), ce qui ne devrait pas exister d'après les spécifications du type de la classe CREATED. En tout, il y a neuf classes de relations pour lesquelles le type spécifié dans la table des relations et les instances du jeu d'entraînement divergent. Dans ce travail, on a supposé que le jeu d'entraînement devrait présenter les mêmes caractéristiques que le jeu de test, et il a donc été décidé d'utiliser les types des classes de relations tels qu'ils apparaissent dans le jeu d'entraînement. On note également que certaines classes de relations ont un type qui est plus restreint en pratique que celui indiqué dans la table des relations fournie sur la page du défi Kaggle. C'est le cas de CREATED, pour laquelle l'objet ne peut pas être de type GROUP_OF_INDIVIDUALS, qui est l'un des sous-types majeurs de la classe ORGANIZATION. De même, les classes WAS_DISSOLVED_IN, WAS_CREATED_IN, et OPERATES_IN n'acceptent pas ce type d'entité en sujet dans le jeu d'entraînement. Restreindre les types, en excluant les paires ayant un objet ou un sujet de type

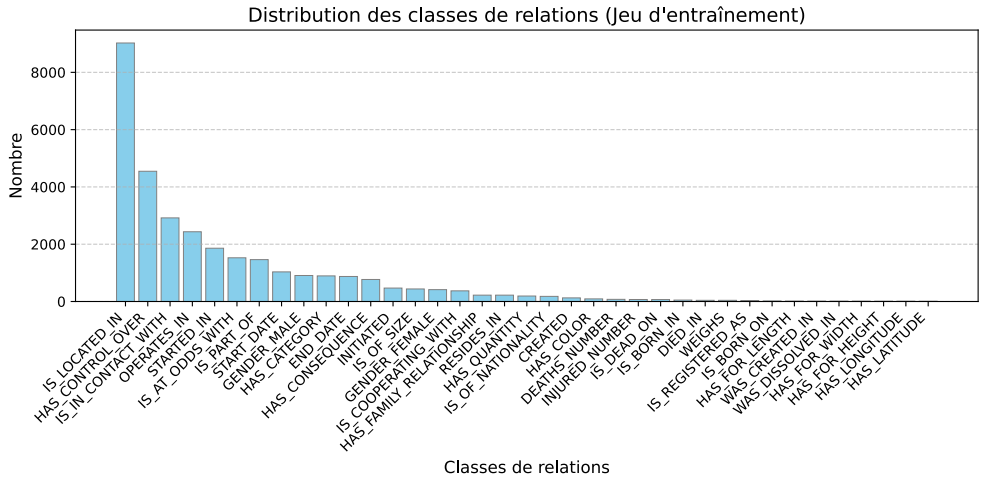


FIG. 1 – *Distribution du nombre d'instances par classes de relations dans le jeu d'entraînement.*

GROUP_OF_INDIVIDUALS pour ces classes de relations, permet alors à la fois de réduire le temps de calcul au niveau de l'entraînement et de l'inférence, tout en augmentant la précision de l'extraction des relations, car cela limite les faux positifs éventuels.

En plus des types des classes de relations, tous les types d'entités, d'attributs, et de relations sont fournis avec une définition. Néanmoins, comme cela sera discuté dans la section 6, les définitions fournies ne suffisent pas tout le temps à annoter avec confiance les différentes relations.

3.2 Distribution des données

La distribution des données présente de nombreux déséquilibres. Ainsi, la distribution du nombre de relations par classe de relations est donnée dans la figure 1 et montre que, si de nombreuses classes ont plus de 1000 instances (huit telles classes), à l'extrême opposé, de nombreuses en ont moins de 100 (16 telles classes), voire moins de 20 (huit telles classes). Aussi, on note que la relation la plus présente, IS_LOCATED_IN, a plus de deux fois plus d'instances que la deuxième plus fréquente, soulignant encore une fois les forts déséquilibres existant dans le jeu de données. D'autres déséquilibres existant dans la distribution des données concernent le nombre d'instances pour chaque type d'entité, ainsi que le nombre de relations d'un même type par exemple de données. Entre autres, certaines classes de relations ont de l'ordre de six instances par texte en moyenne (e.g., HAS_CONTROL_OVER) , tandis que d'autres n'en ont au plus qu'une (e.g., HAS_LATITUDE).

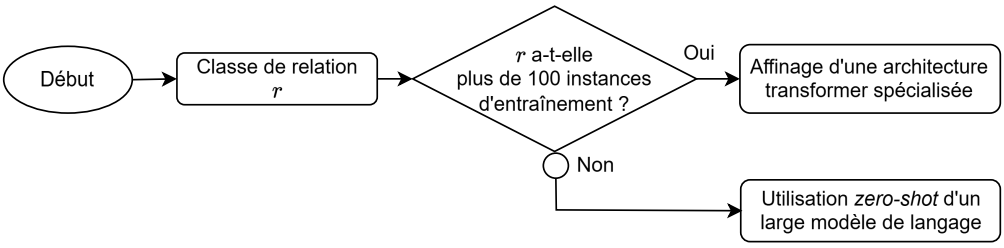


FIG. 2 – *Méthodologie générale.*

3.3 Classes rares vs non-rares

Une distinction est faite entre deux sortes de classes de relations, parmi les 37 présentes dans le jeu de données : les classes rares et les classes non-rares. De manière arbitraire, les classes rares sont celles qui ont au plus 100 exemples, et les classes non-rares celles qui en ont plus de 100. Selon cette définition, les classes non-rares sont celles qui ont au moins 20 exemples de données de validation lors d'une validation croisée à 5 plis. Cela a un intérêt pour réduire l'incertitude sur les scores obtenus sur le jeu d'entraînement lors de la validation des modèles. Les classes rares sont essentiellement des classes de relations dont le type de l'entité objet est un attribut (e.g., LENGTH, HEIGHT, LATITUDE, etc.).

4 Méthodologie

La méthodologie développée pour traiter le défi TextMine 2025 a consisté essentiellement en les blocs suivants, résumés sur la figure 2. Dans un premier temps, l'analyse exploratoire des données révèle si une classe de relation est rare ou non-rare. En fonction de la rareté d'une classe de relation, la méthode de classification et de validation est sélectionnée. Pour les classes rares, une approche sans apprentissage supervisé et utilisant des LLMs pré-entraînés est employée, tandis que les classes non-rares bénéficient de l'affinage d'un modèle transformer sur le jeu d'entraînement, en validation croisée à cinq plis. Pour les deux méthodes, toutes les paires d'entités sujet-objet dont le type correspond au type de la relation sous étude, sont utilisées pour entraînement et validation. Une attention particulière est portée au fait que la séparation des données entre jeu d'entraînement et jeu de validation est faite sur les identifiants des exemples de données, ainsi aucune relation dont le texte est présent dans le jeu d'entraînement ne sera utilisée dans le jeu de validation.

4.1 Pré-traitement sur les types de relation

Dans un premier temps, indépendamment de la méthode employée, pour améliorer les performances, en particulier la précision, et pour réduire la complexité algorithmique, autant lors de l'entraînement que lors de l'inférence, une attention particulière est portée sur le type des relations. En effet, comme précisé dans la section 3.1, chaque classe de relation a un type bien précis qui définit quels types d'entités l'on peut trouver en tant que sujet et en tant qu'objet.

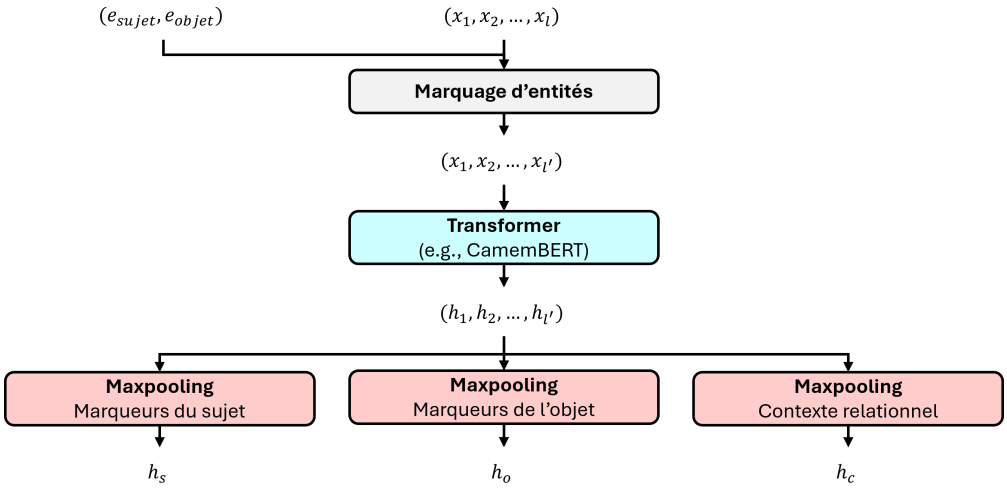


FIG. 3 – *Méthode supervisée : obtention des représentations à l'aide d'un transformer.*

Lors de l'entraînement, toutes les paires d'entités ayant le bon type de la classe de relation sous étude, et se trouvant dans un texte ayant au moins une instance de cette classe de relation sont utilisées. Celles qui sont effectivement reliées par cette relation jouent le rôle d'exemples positifs, tandis que toutes les autres, qui ne sont pas reliées par cette relation, jouent le rôle d'exemples négatifs. Lors de l'inférence, toutes les paires d'entités ayant le bon type sont fournies à la méthode de prédiction employée et le modèle prédit s'il s'agit d'une paire positive ou négative.

Le choix de ne pas inclure, lors de l'entraînement, les paires d'entités issues de textes sans instance de la classe de relation peut sembler contre-intuitif, car, à l'inférence, toutes les paires doivent être testées, y compris celles provenant de tels textes. Cependant, cette approche se justifie par l'objectif d'éviter que les caractéristiques textuelles des données sans relation soient excessivement pénalisées.

4.2 Modèles de classification

Comme indiqué au début de la section 4, la méthode employée dépend de la rareté de la classe. Les classes non-rares sont traitées par la méthode supervisée, tandis que les classes rares sont traitées par la méthode sans entraînement. Dans tous les cas, la démarche est de classifier les paires d'entités ayant un type compatible à la classe de relation sous étude, pour indiquer binairement si elles sont reliées par cette relation. Un modèle est fait pour chaque classe.

4.2.1 Méthode supervisée

Les méthodes employant des architectures transformers étant à la pointe en traitement du langage naturel, l'utilisation d'une architecture transformer pour classifier les paires d'entité

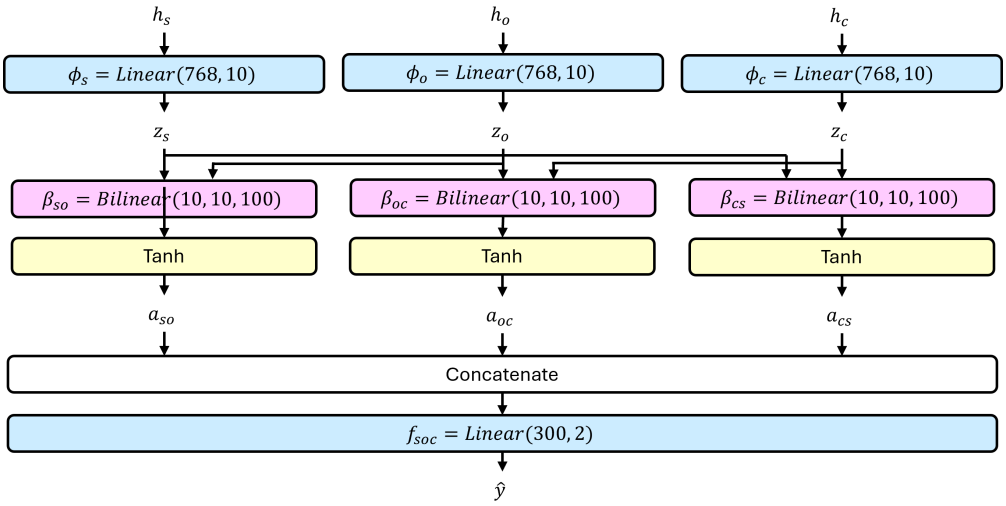


FIG. 4 – *Méthode supervisée : architecture de la classification du triplet de représentations.*

a été choisie. Ici, l'approche s'inspire principalement de Eberts et Ulges (2021) et Zhou et al. (2020). L'idée est de disposer de marqueurs d'entités, permettant de récupérer une représentation contextualisée du sujet et une de l'objet, ainsi qu'une représentation contextualisée du contexte relationnel entre le sujet et l'objet, puis de les utiliser pour classer la paire d'entités en relation ou non-relation. Pour enrichir les caractéristiques utilisées pour la classification, nous étendons les travaux antérieurs en prenant en compte les interactions deux à deux entre sujet, objet et contexte.

Etant donné un document $d = (x_t)_{t=1}^{l}$ et une paire d'entités (e_{sujet}, e_{objet}), la position des mentions d'entités est marquée à l'aide d'un caractère spécial qui indique s'il s'agit de l'entité sujet ou de l'entité objet, ainsi que le type de l'entité (e.g., <S-PLACE> pour une mention du sujet, dans le cas où le sujet est de type PLACE), et est placé juste avant chaque mention. On passe ensuite le document marqué dans un modèle pré-entraîné de langage pour obtenir des représentations contextualisées : $(h_1, h_2, \ldots, h_{l'}) = BERT(x_1, x_2, \ldots, x_{l'})$. Les h_t sont des vecteurs de dimension latente 768 typiquement, dans le cas de l'utilisation d'une architecture BERT (Devlin et al. (2019)). Ensuite, la représentation contextualisée d'une mention d'une entité est la représentation obtenue par passage du marqueur de la mention à travers le transformer. Une entité donnée a parfois plusieurs mentions. Pour prendre en compte toutes les mentions d'une entité, la représentation de l'entité est obtenue par maxpooling sur les représentations des mentions de l'entité, suivant Eberts et Ulges (2021). Ainsi, pour chaque paire d'entités, on obtient une représentation contextualisée h_s du sujet et une h_o de l'objet. De plus, une représentation contextualisée h_c du contexte relationnel est obtenue par maxpooling des représentations des tokens se trouvant entre la première et la dernière mentions de la paire d'entités. On obtient ainsi un triplet de représentations : (h_s, h_o, h_c). Ces étapes sont représentées dans la figure 3.

Les trois représentations obtenues dans un premier temps sont ensuite utilisées pour faire

```
f"""On définit la relation '{r}' comme étant '{RELATIONS_DEFINITIONS[r]}'.

Voici un rapport de renseignement que l'on souhaite étudier : {text}

Dans ce rapport de renseignement, l'entité sujet [{es_all['mentions'][0]}], de type
{es_all['type']}, et l'entité objet [{eo_all['mentions'][0]}, de type {eo_all['type']}
sont-elles reliées par la relation '{r}' ? Réponds simplement Oui ou Non."""
```

FIG. 5 – *Format du prompt.*

la classification de relation, suivant le schéma de la figure 4 et tel que décrit ci-après. Chacune de ces représentations est projetée dans un espace de plus petite dimension : $z_s = \phi_s(h_s)$, $z_o = \phi_o(h_o)$ et $z_c = \phi_c(h_c)$, avec ϕ_s, ϕ_o et ϕ_c des couches linéaires. Puis, les interactions entre le sujet, l'objet, et le contexte sont calculées : $z_{so} = \beta_{so}(z_s, z_o)$, $z_{oc} = \beta_{oc}(z_o, z_c)$, et $z_{cs} = \beta_{cs}(z_c, z_s)$, avec β_{so}, β_{oc} et β_{cs} des couches bilinéaires. Finalement, la fonction d'activation tangente hyperbolique est appliquée sur chacune de ces interactions pour générer de la non-linéarité : $a_{so} = \tanh(z_{so})$, $a_{oc} = \tanh(z_{oc}$, et $a_{cs} = \tanh(z_{cs}$. Enfin, ces activations sont agrégées par concaténation : $a_{soc} = concatenate(a_{so}, a_{oc}, a_{cs})$. Finalement, la classification binaire est réalisée à l'aide d'une tête linéaire sur cette représentation finale : $\hat{y} = f_{soc}(a_{soc})$ avec f_{soc} une couche linéaire avec une sortie à deux dimensions.

4.2.2 Méthode sans entraînement

La méthode sans entraînement consiste en la définition d'un prompt et le traitement de la réponse d'un LLM pour définir si oui, ou non, une relation a lieu entre deux entités données. Cette approche est conceptuellement et dans la réalisation la plus simple, dès lors que l'on dispose de bons LLMs. Le format utilisé pour le prompt donné au LLM est fourni dans la figure 5. L'étiquette r de la classe de relation est donnée, avec sa définition, pour permettre au LLM de comprendre de quelle relation il s'agit. Ensuite, le texte du rapport de renseignement est donné. Puis, la première mention de l'entité sujet et la première de l'entité objet sont fournies, ainsi que leur type d'entité. Une fois ces informations données en contexte, il est demandé au LLM de répondre par "Oui" ou par "Non" afin d'indiquer si les deux entités sont connectées par la relation proposée. Les réponses du LLM sont automatiquement traitées pour détecter si la réponse commence par "Oui" ou par "Non", et classifier selon cela. Utiliser la méthode sans entraînement présente des avantages et des inconvénients. Les avantages sont principalement (1) l'utilisation de LLMs qui sont puissants sur les tâches de compréhension du langage, (2) la disparition de la phase d'entraînement, (3) la facilité à mettre en place, modifier et comprendre le code, et (4) la possibilité d'utiliser toutes les données d'entraînement en tant que validation. Toutefois, les inconvénients sont (1) le temps d'inférence élevé, (2) le coût d'utilisation important lors de l'utilisation de l'API de modèles propriétaires, et (3) l'absence d'entraînement supervisé pour apprendre des règles spécifiques au jeu de données.

Le choix de la première mention de chaque entité est arbitraire et repose sur cette observation : souvent, la première mention d'une entité donne le plus d'informations sémantiques sur cette entité, tandis que les suivantes seront des pronoms reprenant cette entité. Ce n'est néanmoins pas toujours le cas, ce qui limite cette approche. Une autre approche aurait été d'utiliser

toutes les mentions. Notons que la méthode sans entraînement étant utilisée avec des classes rares, il y a peu de données de validation et donc un risque accru de surapprentissage. Ainsi, pour limiter ce risque, il a été décidé de limiter le nombre de stratégies de prompting différentes testées, pour ne pas surapprendre l'ensemble d'entraînement.

5 Résultats

Dans le Défi TextMine 2025, l'on mesure le F1-score pour la classification de chacune des 37 classes de relations, et le score final est calculé comme étant le F1-score macro. Ainsi, les contributions des classes sont équipondérées, qu'importe le nombre d'instances de chaque classe. Il convient donc d'accorder la même importance à la classification de chacune des 37 relations.

5.1 Expériences

Pour la méthode supervisée, le modèle transformer utilisé est un CamemBERT pré-entraîné (Martin et al. (2020a)), car ce modèle est spécialisé pour traiter le langage français. Toutes les couches du modèle sont affinées. L'entraînement est effectué avec une taille de lot de 16, un taux d'apprentissage valant au plus 10^{-5}, et un planificateur utilisant une décroissance en cosinus. Le nombre maximal d'époques est fixé à 20, avec une régularisation de poids de 0,01. La fonction de perte utilisée est l'entropie croisée, et l'optimiseur AdamW est appliqué. Un mécanisme d'arrêt anticipé surveille le F1-score avec une patience de 5 époques. De plus, pour prendre en compte le déséquilibre entre le nombre d'exemples positifs et d'exemples négatifs, un échantillonneur aléatoire et équilibré est utilisé pour avoir une proportion égale d'exemples positifs et d'exemples négatifs dans chaque lot de données lors de l'entraînement.

Pour la méthode sans entraînement, des modèles propriétaires ainsi que des modèles en libre accès ont été testés, parmi lesquels essentiellement GPT-4o (OpenAI (2024b)) et GPT-4o-mini (OpenAI (2024)), ainsi que Gemma2-9B (Gemma Team (2024)). Les modèles propriétaires cités performant mieux, ils ont été sélectionnés. GPT-4o est utilisé pour les relations traitant de latitude, de longitude, de taille, de longueur, et de poids, car ces relations ont très peu de paires d'entités du bon type, ce qui permet d'utiliser le modèle sans payer un coût excessif. Autrement, GPT-4o-mini est utilisé pour les autres classes rares, via l'API de OpenAI.

5.2 Scores

Finalement, on obtient sur le jeu d'entraînement un résultat de 65,6% de F1-score macro, sur le jeu de test public 63,2%, et sur le jeu de test privé 64,0%. L'écart entre l'entraînement et le test suggère un possible surapprentissage. N'ayant cependant pas accès au détail du score pour chaque classe de relation sur le jeu de test, il est difficile de diagnostiquer exactement la raison de cet écart. Les F1-scores obtenus sur le jeu d'entraînement sont indiqués dans la figure 6. Lors de la validation des modèles, les performances diffèrent sensiblement d'une classe de relation à l'autre. On peut segmenter les classes en cinq catégories disjointes : les classes très faciles (80-100%), les plutôt faciles (65-80%), les moyennement faciles (50-65%), les plutôt difficiles (40-50%), et les très difficiles (0-40%).

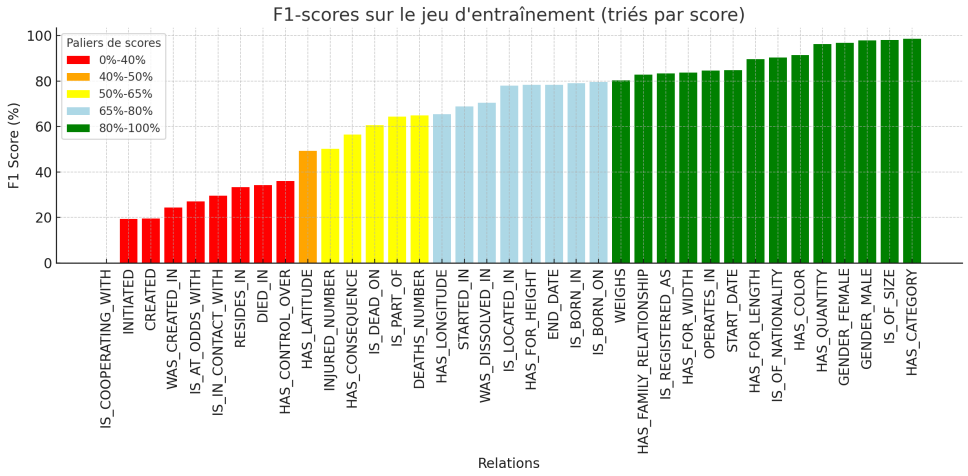


FIG. 6 – *F1-scores sur le jeu d'entraînement.*

Les classes très faciles à prédire sont uniquement des relations reliant une entité à un attribut, telles que GENDER_MALE ou HAS_CATEGORY. Les classes plutôt faciles relient des entités soit à un attribut, soit à un lieu, soit à une date. En revanche, les classes moyennes et plus difficiles impliquent essentiellement des organisations et des événements. La difficulté à prédire ces classes peut être due à la diversité des types d'entités impliquées dans ces relations, induisant une complexité accrue pour l'apprentissage de ces relations. Certaines classes comme HAS_CATEGORY peuvent être très facilement prédites, ce qui est encourageant. Cependant, d'autres relations restent récalcitrantes. En particulier, la classe IS_COOPERATING_WITH présente une difficulté majeure car, qu'importe les méthodes employées, elle dépasse péniblement 10% de F1-score sur le jeu d'entraînement. Une partie de l'explication de cette difficulté est que, lors de l'entraînement, seulement 1% des données d'entraînement pour cette classe sont des exemples positifs, contre 99% d'exemples négatifs. En général, les classes ayant un fort déséquilibre entre exemples positifs et exemples négatifs obtiennent de faibles scores. C'est le cas également de CREATED, avec 0,3% d'exemples positifs et qui est une classe très difficile.

Une étude ablative a révélé qu'utiliser uniquement l'approche avec entraînement fournit un F1-score macro de 34.1% sur l'ensemble des classes rares dans le jeu d'entraînement, contre 64.1% pour la méthode sans entraînement. Sur ces classes rares, si l'on prend, pour chaque classe, la meilleure méthode au cas par cas, l'on améliore encore le F1-score à 67.8%. En effet, IS_DEAD_ON, IS_BORN_IN et HAS_COLOR préfèrent la méthode avec apprentissage. De plus, en prenant la meilleure méthode sur les classes rares, et la méthode supervisée sur les classes non-rares, l'on obtient 65.6% de F1-score sur l'ensemble, contre 51.0% avec l'usage unique de la méthode supervisée, améliorant ainsi sensiblement les performances. Pour des raisons de coût, l'étude ablative a été menée uniquement sur les classes rares. Aussi, des tests sur certaines classes ont montré que la méthode sans entraînement améliorait difficilement des classes non-rares, avec davantage d'instances labellisées.

FIG. 7 – *Nuages de mots différentiels, à gauche présentant les termes présents plus souvent lorsqu'il n'y a pas de relation DIED_IN et à droite lorsqu'il y a une relation DIED_IN.*

6 Discussion des données synthétiques

6.1 Relation HAS_LATITUDE

En inspectant à la main les prédictions de la méthode sans entraînement pour la relation HAS_LATITUDE, il est apparu que les données peuvent manquer soit de cohérence dans les annotations, soit de spécifications dans les définitions des relations. Par exemple, le texte 4964 du jeu d'entraînement comporte la phrase *"Il a été arrêté à son domicile situé à Hong Kong, à la latitude 22° 19' 0.0048" N, le 13 décembre 2016."* et le seul lieu relié à une latitude est *"Hong Kong"*, tandis que *"son domicile* n'a pas de relation HAS_LATITUDE. À l'inverse, dans le texte 4924, la phrase *"Une équipe de la police dirigée par Martin Deneuve a appréhendé les braqueurs qui se cachaient dans une maison abandonnée localisée à Reims (Latitude 49.258329)."* relie *"Maison abandonnée"* à la latitude donnée, mais pas *"Reims"*. Ces variations soulèvent deux questions : pourquoi ne pas relier la latitude à la fois à la ville et au lieu spécifique ? Et pourquoi, selon les cas, relier l'un ou l'autre ? Ces observations montrent que des règles plus détaillées sur les relations permettraient sans doute une prédiction plus juste des relations. En particulier, avec la méthode employée dans ce travail, GPT-4o avait tendance à prédire tous les lieux comme attachés à la latitude, obtenant 100% de rappel, mais se retrouvant pénalisé au niveau de la précision. L'annotation de ces données par plusieurs annotateurs et le calcul d'un score d'accord inter-annotateurs permettrait d'établir une performance maximale atteignable par un modèle d'apprentissage automatique.

Enfin, la qualité rédactionnelle des textes du jeu d'entraînement suggérait que ces données synthétiques avaient été générées en utilisant un large modèle de langage, ce qui est corroboré à la lecture de Giordano et al. (2024). Dès lors, nous avons cherché à regarder plus finement les termes et structures présentes dans les textes du jeu de données d'entraînement, pour trouver d'éventuels motifs spécifiques au LLM employé lors de la génération des données.

6.2 Relation DIED_IN

Une étude de cas a été menée sur la relation DIED_IN, l'une des plus difficiles à prédire, malgré l'intuition que la notion de décès semble facile à identifier dans un texte. Cette relation relie un sujet de type ACTOR à un objet de type EVENT, comme un individu décédant lors d'un événement. Par ailleurs, le type EVENT inclut le sous-type SUICIDE qui, par na-

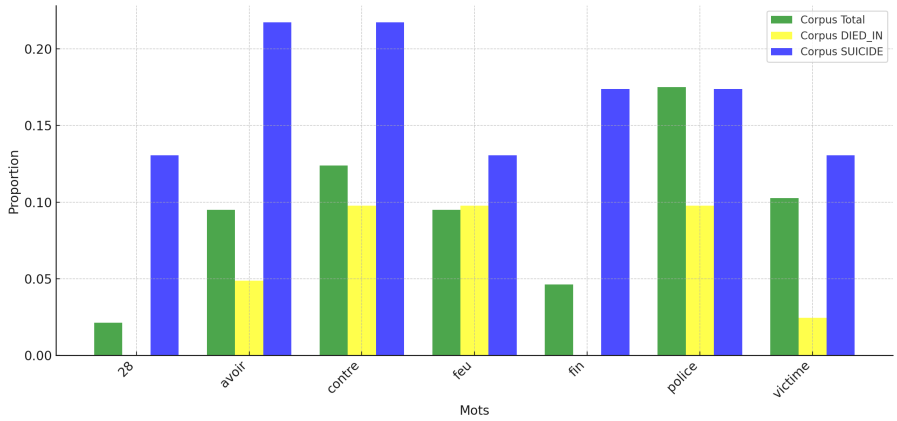


FIG. 8 – *Proportions de textes contenant les mots "28", "avoir", "contre", "feu", "fin", "police", et "victime", dans trois corpus différents.*

ture, suggère sans ambiguïté un décès. Une règle que l'on peut alors supposer est : tout texte contenant une entité SUICIDE doit contenir une relation DIED_IN. Or, dans le jeu d'entraînement, on observe que sur 23 textes possédant une entité SUICIDE, seulement 5 (22%) ont une relation DIED_IN. Cette observation surprenante permet de nuancer les faibles performances sur la classe DIED_IN et interroge la qualité des annotations. À la lecture, presque tous les textes avec SUICIDE et sans relation DIED_IN pourraient inclure cette relation, à l'exception notable du texte 3901, qui traite d'une association de prévention contre le suicide et d'une tentative sans décès. Pour illustrer, un exemple d'incohérence suspectée dans les annotations de DIED_IN et SUICIDE concerne les textes 41634 et 41922, où le même personnage fictif "Joshua Clavier" est témoin d'un suicide dans chaque texte. Dans le premier cas, DIED_IN est bien indiquée, tandis que dans le second cas cette relation n'apparaît pas.

Suspectant que l'utilisation de LLMs pour générer le jeu de données ait induit des corrélations fallacieuses entre certains termes, les entités SUICIDE, et les relations DIED_IN, nous avons visualisé, à l'aide d'un nuage de mots (voir figure 7, les termes les plus présents différentiellement entre les textes contenant une entité SUICIDE avec ou sans relation DIED_IN. Le nuage de droite est peu exploitable car seuls cinq textes ont une entité SUICIDE et une relation DIED_IN. Le nuage de gauche, en revanche, révèle que des termes comme "police", "victime", "fin", "feu", ainsi que des mots inattendus comme "avoir", "contre", ou "28", pourraient être corrélés à l'absence de relation DIED_IN. En allant plus loin, il apparaît que ces termes ne sont dans aucun des textes ayant à la fois une relation DIED_IN et une entité SUICIDE. Par ailleurs, la figure 8 montre, pour ces termes, la proportion de textes les contenant parmi le corpus global (800 textes), les 41 textes avec une relation DIED_IN, et les 23 textes avec une entité SUICIDE. On observe que ces termes ciblés sont significativement plus présents dans les textes avec SUICIDE que dans l'ensemble du corpus, et moins présents dans les textes avec DIED_IN. Ces statistiques ne parlant que de corrélation, tirer des conclusions est difficile, mais elles suggèrent l'existence de "portes dérobées" exploitables pour améliorer l'extraction de relations dans un jeu de données synthétiques. Par exemple, aucun des 37 textes contenant le mot "fin" n'a de relation DIED_IN.

7 Conclusion

Combiner une approche supervisée affinant des transformers modélisant les interactions entre sujet, objet et contexte, avec une méthode sans entraînement reposant sur des LLMs a atteint 64,0% de F1-score sur le Défi TextMine 2025 d'extraction des relations. L'enjeu majeur de validation des modèles a incité à différencier les méthodes selon la rareté des relations. Enfin, l'étude des données synthétiques suggère le recours à plus de spécifications sur l'annotation des relations, ainsi que le calcul d'un score d'accord entre annotateurs pour estimer le score maximal atteignable pour un modèle d'apprentissage sur ce jeu de données.

Références

Prieur, M., G. Gadek, A. Guille, H. M. Rawsthorne, P. Cuxac, et C. Lopez (2025). Défi TextMine'25 - Extraction de relations pour analyser des rapports de renseignement. In *Actes de l'atelier TextMine'25*, Extraction et Gestion des Connaissances 2025 (EGC'25), pp. à paraître.

Giordano, B., M. Prieur, N. Vuth, S. Verdy, K. Couzot, G. Serasset, G. Gadek, D. Schwab, et C. Lopez (2024). POPCORN : Fictional and Synthetic Intelligence Reports for Named Entity Recognition and Relation Extraction Tasks. In *28th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2024)*, 28th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2024), Seville, Spain. Elsevier.

OpenAI (2024b). Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`.

OpenAI (2024). Gpt-4o mini : advancing cost-efficient intelligence. `https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/`.

Gemma Team (2024). Gemma. doi:10.34740/KAGGLE/M/3301.

Eberts, M. et A. Ulges (2021). Span-based joint entity and relation extraction with transformer pre-training. doi:10.3233/FAIA200321.

Zhou, W., K. Huang, T. Ma, et J. Huang (2020). Document-Level Relation Extraction with Adaptive Thresholding and Localized Context Pooling. doi:10.48550/arXiv.2010.11304.

Martin, L., B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, E. V. d. l. Clergerie, D. Seddah, et B. Sagot (2020a). CamemBERT : a Tasty French Language Model. doi:10.48550/arXiv.1911.03894.

Devlin, J., M.-W. Chang, K. Lee, et K. Toutanova (2019). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. doi:10.48550/arXiv.1810.04805.

Summary

For the TextMine 2025 challenge on relation extraction from synthetic reports, we propose a hybrid approach combining large language models and pre-trained language models for relation classification on text segments, achieving a 64.0% F1-score on the private test set.

Défi TextMine 2025 : Utilisation des Grands Modèles de Langue pour l'Extraction de Relations dans les Rapports de Renseignement

Mohamed Ettaleb*, Mouna Kamel*,**
Véronique Moriceau*, Nathalie Aussenac-Gilles*

*IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Toulouse
**Espace-Dev, Université de Perpignan

Introduction

L'extraction de relations (RE) vise à identifier et caractériser les relations sémantiques entre des entités dans un texte, une tâche clé en traitement automatique du langage naturel (TALN). Les approches supervisées traditionnelles reposent sur l'annotation des entités suivie de la prédiction des relations entre elles. Récemment, les méthodes séquence-à-séquence ont simplifié ce processus en générant directement les relations sous forme de chaînes cibles. Les grands modèles de langage (LLMs) se distinguent par leur capacité à traiter efficacement ces tâches complexes. Dans le cadre du défi TextMine 2025, l'objectif est d'automatiser la tâche d'extraction de relations à partir de rapports complexes pour le renseignement et la défense.
Ce défi offre une opportunité unique d'évaluer les performances des LLMs dans des scénarios réalistes.
Nous proposons une approche utilisant le modèle Llama3 (**?**) pour détecter et classer les relations entre paires d'entités dans un texte. Nous combinons la puissance des LLMs avec des étapes de filtrage préalable reposant sur les types d'entités et de relations. L'objectif est d'évaluer dans quelle mesure un LLM peut répondre aux besoins spécifiques de l'extraction de relations dans des contextes complexes, tout en mettant en lumière leurs limites et les défis à surmonter.

Approche proposée pour l'extraction des relations avec Llama3

Pour extraire toutes les relations possibles entre les paires d'entités dans un texte, notre méthode se décompose en plusieurs étapes décrites ci-dessous.
1. Génération de toutes les combinaisons de paires d'entités : Pour chaque texte, toutes les entités mentionnées sont identifiées et combinées en paires. Si l'ensemble d'entités est représenté par $E = \{e_1, e_2, e_3, \ldots, e_m\}$, les combinaisons générées incluent toutes les paires possibles, y compris les paires auto-référentielles, telles que (e_1, e_2), (e_1, e_3), jusqu'à (e_m, e_m). Cette étape garantit que toutes les interactions potentielles entre les entités sont couvertes.
2. Filtrage des paires selon les types d'entités et relations possibles : Les paires générées sont ensuite filtrées en utilisant des définitions des relations possibles données par le guide d'annotation, stockées dans un dictionnaire appelé `Relations_Definition`. Ce

dictionnaire associe des relations spécifiques à des types d'entités, par exemple : une entité de type **Actor** peut être en relation *Is_Located_In* avec une entité de type **Place**. Chaque paire est vérifiée pour s'assurer que les types des entités correspondent à une relation valide dans `Relations_Definition`. Si une paire respecte ces contraintes, elle est conservée ; sinon, elle est écartée. Ainsi, seules les paires d'entités valides restent dans la liste finale.
3. Identification des relations potentielles entre entités : Pour chaque paire d'entités retenue après filtrage, la paire est vérifiée dans le jeu de données annotées (gold standard). Si une relation annotée existe entre les deux entités, elle est associée à la paire. Sinon, l'étiquette `PAS_DE_RELATION` est attribuée.
4. Génération du prompt : Un prompt personnalisé est généré pour chaque paire. Ce prompt inclut le texte contenant les entités, les types des entités, et une liste des relations possibles selon les types des entités (incluant toujours `PAS_DE_RELATION`). La sortie attendue pour ce prompt est la relation correcte, si elle est présente, ou `PAS_DE_RELATION`.
5. Entraînement du LLM : Les prompts générés sont utilisés pour fine-tuner le modèle Llama3. Chaque prompt comprend comme entrée le contexte du texte et les informations sur les entités. La sortie attendue est la relation correspondante ou `PAS_DE_RELATION`. Ce processus d'entraînement permet au modèle de comprendre les relations entre les entités en fonction de leur contexte et des définitions disponibles dans `Relations_Definition`.

Résultats et discussion

Nous avons divisé le jeu de données annotées, composé de 1200 textes, en 600 textes pour l'entraînement de Llama3, 200 pour la validation et 400 pour le test. Le corpus de validation a été constitué en respectant la même distribution des classes que dans le corpus d'entraînement afin d'assurer une évaluation équitable des performances.
Le modèle a obtenu un score F1 macro de **0,61** sur le jeu de validation et de **0,38** sur le jeu de test, mettant en évidence des difficultés à généraliser certaines relations. Parmi les relations bien détectées sur le jeu de validation figurent *HAS_CONSEQUENCE* (0,94), *HAS_QUANTITY* (0,86), *IS_OF_NATIONALITY* (0,93), *IS_OF_SIZE* (0,95) et *GENDER_FEMALE* (0,93). En revanche des relations comme *WAS_CREATED_IN* (7 occurrences), *DIED_IN*(15 occurrences) et *IS_BORN_IN*(15 occurrences) ont obtenu des scores de F1 très faibles (entre 0 et 0,15), ce qui s'explique par leur faible fréquence dans le corpus d'entraînement. Un autre facteur ayant impacté les performances est la qualité des annotations. De nombreuses annotations contenaient des erreurs, comme nous l'avons constaté sur un sous-ensemble du corpus de validation. Par exemple, une relation *Is_Located_In* était annotée entre "volés" (verbe) et "Lisbonne" (nom), ce qui complique l'interprétation sémantique. Une mention plus explicite, telle que "articles volés" au lieu de "volés", aurait permis de mieux refléter la relation et d'améliorer la qualité de l'entraînement. Ces erreurs ont probablement perturbé l'apprentissage du modèle.
Finalement, même les LLM tels que Llama3 rencontrent des difficultés pour généraliser lorsque les données présentent un déséquilibre important entre les classes de relations.

1 Acknowledgement

This work was carried out within the ECLADATTA project funded by the French National Research Agency under grant ANR-22-CE23-0020.

Participation de l'équipe Défense au défi TextMine'25 en extraction de relations dans des bulletins de renseignement

Nicolas Diniz*,
Nihel Kooli*,
Lucie Chasseur**,
Pauline Soutrenon**

*prénom.nom@intradef.gouv.fr,
**prénom.nom@inria.fr

Résumé. L'extraction d'informations, en particulier l'extraction de relations dans un contexte spécifique, reste un défi malgré les progrès réalisés dans le domaine du Traitement Automatique des Langues (TAL). Cet article s'inscrit dans le cadre du défi TextMine'25 [1] , proposé lors de la conférence EGC 2025, qui porte sur l'extraction de relations entre entités et événements d'intérêt militaire dans des rapports de renseignement simulés. Nous y présentons une analyse approfondie du corpus utilisé, ainsi qu'une méthode d'augmentation des données basée sur des modèles de langage de grande taille. Cette approche, basée sur le transformeur pré-appris multilingue mDeBERTa, vise à enrichir les annotations et diversifier les exemples pour l'entraînement des modèles.

1 Introduction

L'extraction d'informations, en particulier la tâche d'extraction de relations d'intérêt dans des contextes spécifiques, représente un défi persistant dans le domaine du Traitement Automatique des Langues (TAL). Bien que les récents progrès en réseaux neuronaux profonds, tels que BERT (Devlin et al., 2019) ou GPT (Brown et al., 2020), aient considérablement amélioré les performances des modèles sur diverses tâches linguistiques, leur application à des domaines spécialisés reste complexe. La sémantique des relations et la précision contextuelle requises dans ces contextes soulèvent encore des questions non résolues.

Parmi les principaux obstacles figurent l'interprétation de textes ambigus, l'identification de relations implicites, et la gestion de données limitées, souvent observées dans des domaines sensibles comme le renseignement militaire. Dans ces contextes, où la qualité et la contextualisation des informations extraites sont cruciales, des avancées méthodologiques spécifiques s'avèrent nécessaires.

1. https://www.kaggle.com/competitions/defi-text-mine-2025/overview

Le défi TextMine'25, organisé dans le cadre de la conférence Extraction et Gestion des Connaissances (EGC) 2025, illustre parfaitement ces problématiques. Il vise à évaluer des approches d'extraction de relations entre entités et événements dans des rapports de renseignement factices. Ce défi, à l'intersection du TAL et du renseignement d'intéret militaire, met en lumière l'importance de modèles capables de s'adapter aux particularités des textes spécialisés tout en garantissant une précision optimale.

Pour relever ces enjeux, la disponibilité d'ensembles de données représentatifs et annotés est essentielle. L'émergence des modèles de langage de grande taille (*Large Language Models*, LLM) offre de nouvelles perspectives, notamment pour la génération de données annotées supplémentaires et diversifiées. Ces modèles permettent non seulement d'améliorer l'apprentissage, mais aussi de concevoir des approches innovantes pour renforcer la robustesse et la généralisation des systèmes d'extraction dans des contextes complexes.

Dans cet article, nous analysons en détail le corpus utilisé pour cette tâche et proposons une stratégie d'augmentation de données basée sur les LLM. Nous évaluons ensuite les performances des modèles entraînés avec et sans augmentation de données, en explorant l'impact de cette méthode sur la précision et la diversité des exemples d'entraînement. Les résultats obtenus mettent en lumière les avantages de notre approche et ouvrent des perspectives prometteuses pour des applications dans des domaines spécialisés.

Les codes sources développés dans le cadre du défi sont publiquement disponibles sur github [2].

2 Étude de corpus

2.1 Données à disposition

L'objectif principal du défi était de proposer une méthode d'extraction de relations entre des éléments textuels. Pour mener à bien cette tâche, un *dataset* composé de 800 documents factices a été élaboré et annoté manuellement. Au total, le *dataset* comprend 40 types d'entités distincts, 15 types d'attributs et 37 types de relations.

2.2 Distribution du corpus

Dans un premier temps, nous avons examiné la répartition des entités et des relations dans le corpus d'entraînement. Une observation initiale met en évidence un déséquilibre marqué entre les catégories, en particulier pour la relation *IS_LOCATED_IN*, qui est massivement sur-représentée par rapport aux autres types de relations.

2. `https://github.com/DinizNicolas/defi_textmine_2025`

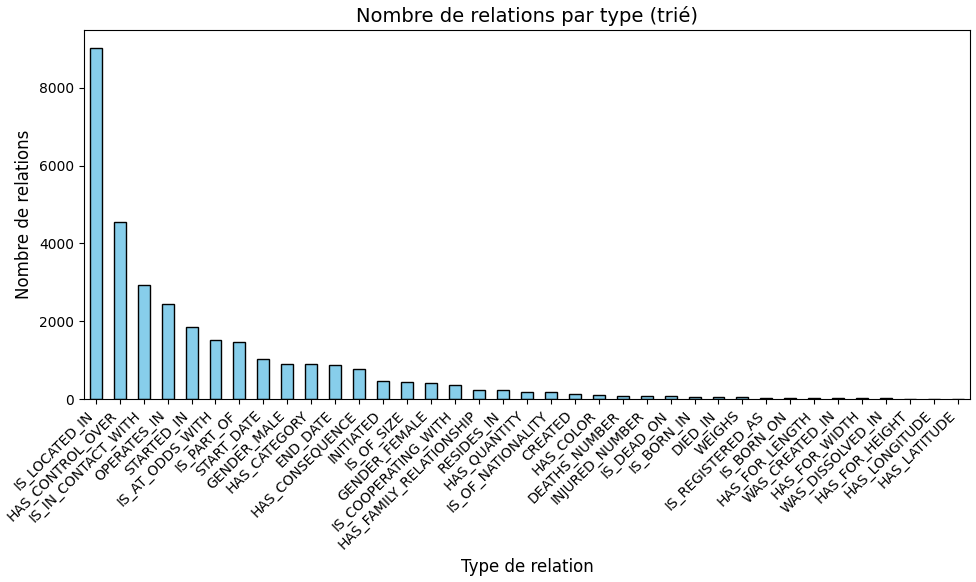


FIG. 1 – *Répartition des relations.*

Cependant, la distribution des relations influence directement la performance des modèles, notamment dans des contextes avec des déséquilibres importants. Une annotation équilibrée et détaillée permet non seulement de former des modèles plus robustes, mais aussi de répondre à des cas spécifiques comme les relations rares ou les tâches inter-dépendantes. (Delaunay et al., 2023) (Bassignana et Plank, 2022)

Afin donc d'évaluer l'impact de ce déséquilibre sur la performance, une première expérimentation a été réalisée sans modification du corpus. Les résultats montrent que les relations ayant moins de 100 occurrences affichent une micro-F1 faible ou nulle, ce qui indique un effet direct du déséquilibre des données sur les performances de ces catégories. De plus, malgré une fréquence élevée de certaines relations, les performances demeurent faibles. Des ajustements dans la conception du corpus et la stratégie de modélisation sont donc nécessaires pour améliorer les performances.

En outre, nous avons pu remarquer que certaines relations, tableau 1 et parfois certaines entités, tableau 2, avaient une annotation différente dans le guide et dans les données ce qui a conduit à une absence de représentation pour ces catégories. Voici une liste non exhaustive de ce que nous avons pu remarquer et donc adapter :

Guide	Données
death number	deaths number
created in	was created in
dissolved in	was dissolved in

TAB. 1 – *Relations différentes entre le guide et les données.*

Guide	Données
hooliganism trouble-making	hooliganism troublemaking
criminal	terrorist or criminal
non governmental organization	non governmental organisation
non military organization	non military organisation
material	materiel
intergovernmental organization	intergovernmental organisation
coup d'état	coup d'etat
military organization	military organisation

TAB. 2 – *Entités différentes entre le guide et les données.*

2.3 Caractéristiques des relations

Les relations sont annotées comme suit : [entité1,relationA,entité2]. On notera le couple d'entités : entité1-entité2. Les entités 1 et 2 pourront être référencé par "entité", "attribut" ou leur label.

Les relations présentent des caractéristiques différentes. Notamment pour les relations entre une entité et un attribut, et les relations entre deux entités. On remarque tableau 6 que les couples entité-entité peuvent avoir plusieurs relations simultanément. Tandis que les couples entité-attribut ne peuvent avoir qu'une seule relation à la fois. Dans la colonne "multiple", lorsqu'il y a 1 ou 2, ce sont des erreurs d'annotations.

Néanmoins, il y a quelques exceptions :
- Pour les couples *EVENT-TIME*, les relations *START_DATE* et *END_DATE* sont possible simultanément.
- Pour les relations *INITIATED*, *HAS_CONSEQUENCE* et *DIED_IN*, il se trouve que le couple entité-EVENT n'a qu'une de ces relations.
- Les deux relations *GENDER* sont entre une entité *PERSON* et elle-même. Dans le corpus un unique genre est possible.

Pour la suite des expérimentations les relations sont séparées en deux groupes. Le groupe des relations de type entité-attribut, avec les deux relations *GENDER*, et excepté les relations *START_DATE* et *END_DATE*. Le reste des relations forment le deuxième groupe. Les relations *INITIATED*, *HAS_CONSEQUENCE* et *DIED_IN* sont dans le deuxième groupe puisqu'il semble préférable de garder les relations autour des *EVENT* dans le même groupe.

3 Augmentation des données à l'aide de LLM

Pour remédier au déséquilibre, nous avons choisi de recourir à l'augmentation de données à l'aide de modèles de langage de grande taille (ou *Large Language Model* en anglais, LLM). En effet, l'intégration de données synthétiques générées par LLM constitue une approche prometteuse pour rééquilibrer les corpus utilisés en extraction de relations sémantiques. Les corpus annotés manuellement présentent souvent des déséquilibres significatifs, avec des relations fréquentes sur-représentées et des relations rares insuffisamment couvertes, ce qui limite la capacité des modèles d'apprentissage à généraliser efficacement. Les LLM peuvent être exploités pour générer des exemples supplémentaires de relations sous-représentées en s'appuyant sur leurs vastes connaissances encodées et leur capacité à produire du texte contextuellement cohérent (Armingaud et al., 2024). Ces données augmentées permettent d'atténuer les biais dans les *datasets* tout en diversifiant les contextes relationnels, renforçant ainsi la robustesse des modèles formés. De plus, l'utilisation de ces approches réduit le coût d'annotation manuel tout en maintenant une qualité satisfaisante des données, à condition d'établir des filtres stricts pour vérifier la validité des exemples générés.

Nous avons donc choisi d'augmenter le nombre d'exemples de relations pour certaines catégories qui présentaient un score nul de micro-F1 après la première exécution. Ces catégories spécifiques sont : *WAS_DISSOLVED_IN*, *WAS_CREATED_IN*, *CREATED*, *DIED_IN*, *IS_COOPERATING_WITH*, et *IS_BORN_IN*.

En générant des exemples supplémentaires pour ces relations sous-représentées, nous visons à améliorer leur couverture dans le corpus et, par conséquent, à renforcer la performance des modèles d'apprentissage sur ces classes.

3.1 Choix du LLM

Nous avons utilisé le modèle Llama 3.2 3B via la plateforme Ollama [3] pour effectuer l'augmentation de données. Ce modèle, caractérisé par sa taille réduite et ses faibles exigences en termes de ressources computationnelles, permet d'obtenir de bonnes performances tout en restant efficace en termes de consommation de mémoire et de puissance de calcul.

3.2 Stratégie de formulation du *prompt*

Différentes configurations de *prompts* ont été testées pour optimiser et stabiliser les résultats. La figure 3 illustre celui retenu, avec les contraintes imposées au modèle décrites ci-après.

Nous introduisons en premier lieu une restriction sur la taille des phrases générées (100 mots maximum) afin de réduire les risques d'erreur du modèle. Cette limitation assure également une meilleure uniformité des données générées et permet de faciliter leur traitement par la suite.

3. `https://github.com/ollama/ollama`

Nous définissons ensuite le format attendu (JSON) avec un identifiant, un texte, des entités et une relation. Nous fournissons un exemple de manière à ce que le modèle puisse reproduire le format attendu.

Enfin, nous précisons que seule la phrase générée, sans explications ni introduction, doit être produite. En l'absence de cette précision dans le *prompt*, le modèle Llama 3.2 a tendance à inclure des explications sur la phrase générée, ce qui complique l'exploitation de ces phrases.

Afin de clarifier la relation attendue, nous fournissons une brève définition au début du *prompt* (par exemple : "la date de naissance d'une personne civile" pour la relation *IS_BORN_IN*). Nous incluons également des explications concernant les annotations attendues à la fin du *prompt*. L'objectif principal de ces explications est de s'assurer que les entités soient effectivement présentes dans le texte et qu'une relation entre elles existe.

Ce *prompt* a été appliqué à l'ensemble des 6 catégories en modifiant à chaque fois la définition de la relation au début du *prompt*, l'exemple JSON et les explications sur l'exemple à la fin du *prompt*. Nous avons opté pour des exemples courts pour chaque catégorie afin de simplifier les consignes au maximum et ainsi limiter les risques d'erreurs.

3.3 Problématiques liées à la génération de texte

Au cours de nos différentes expérimentations sur l'augmentation de données, nous nous sommes confrontés à des problématiques de génération de texte. Parfois, certaines phrases sont redondantes voire même identiques (la même phrase générée sur plus de 1 000 exemples avec les mêmes entités). Il a donc parfois été nécessaire de relancer plusieurs fois le *prompt* et même de redémarrer le modèle.

Malgré les contraintes spécifiées dans le *prompt* (voir la sous-section précédente), les sorties du modèle ne correspondaient pas toujours au format attendu. En effet, nous avons rencontré des problèmes récurrents d'encodage, de format (principalement l'ajout de caractères comme des *backslashs*, des nouvelles lignes, des espaces ou encore des tabulations) et de contenu. Au niveau du contenu, les relations ne portaient pas toujours le label attendu et les entités fournies par le modèle n'étaient pas toujours présentes dans la phrase générée.

Enfin, le modèle semble présenter des limitations quant à la détermination précise des positions de début (*start*) et de fin (*end*) des entités. Les informations générées sont souvent peu fiables, avec des erreurs fréquentes dans les positions indiquées. Ce phénomène est fréquemment observé avec les LLM, qui, bien qu'efficaces pour des tâches linguistiques générales, peinent à gérer des aspects plus structurés et précis du texte, tels que la localisation exacte d'entités.

Pour pallier ces problèmes et éviter l'introduction de bruit dans notre *dataset* d'entraînement, nous avons mis en place un processus de post-traitement des sorties du modèle, que nous décrivons ci-après.

3.4 Post-traitement des réponses du LLM

Les phrases générées par le LLM n'étant pas toujours de qualité constante, nous avons mis en place un post-traitement automatique afin de nettoyer ou d'écarter les sorties générées par le modèle qui comportaient des erreurs. Ce traitement automatique nous a permis de :

— corriger les problèmes d'encodage lorsque cela était nécessaire ;
— vérifier et corriger le format des données (par exemple, s'assurer que la sortie est bien sous format JSON sans espaces ou tabulations additionnels et si ce n'est pas le cas un dictionnaire correctement formé est re-créé automatiquement) ;
— vérifier l'exactitude des données (par exemple, vérifier si les entités fournies par le modèle sont bien présentes dans le texte généré et si ce n'est pas le cas la sortie est écartée) ;
— corriger le label de la relation si celui-ci est traduit dans une autre langue (par exemple, *EST_COOPERANT_AVEC* au lieu de *IS_COPPERATING_WITH*) ;
— corriger les positions de début (*start*) et de fin (*end*) des entités.

Ces différentes opérations permettent d'obtenir des données plus fiables et mieux structurées, facilitant ainsi leur utilisation dans le processus d'entraînement.

3.5 Résultats

Nous présentons dans le tableau 3, les résultats de l'augmentation de données sur les 6 catégories : *WAS_DISSOLVED_IN*, *WAS_CREATED_IN*, *CREATED*, *DIED_IN*, *IS_COOPERATING_WITH*, *IS_BORN_IN*. Malgré le post-traitement automatisé décrit précédemment, nous observons une perte significative des données : seules 37 % des phrases générées par le LLM sont retenues pour l'ajout au *dataset* d'entraînement (soit 3 210 nouvelles phrases avec leur annotation).

Parmi les 6 catégories, seule la catégorie *IS_BORN_IN* semble très bien fonctionner, puisque 84 % des phrases générées ont été retenues.

	Nombre de phrases générées	Nombre de phrases retenues	Pourcentage de phrases retenues
WAS_DISSOLVED_IN	1 582	311	20 %
WAS_CREATED_IN	1 574	418	27 %
CREATED	1 396	393	28 %
DIED_IN	1 600	579	36 %
IS_COOPERATING_WITH	1 571	615	39 %
IS_BORN_IN	1 069	894	84 %
Total	8 792	3 210	37 %

TAB. 3 – *Résultats de l'augmentation de données sur les 6 catégories.*

En outre, nous avons constaté que, malgré un premier post-traitement, certaines données générées restaient incohérentes. Cela a motivé la mise en place d'un nouveau

filtrage plus rigoureux pour garantir la cohérence et la pertinence des données.

Ce processus vise à isoler les relations correspondant à une cible spécifique, par exemple une relation de type *WAS_CREATED_IN*, et à vérifier qu'elles respectent un format attendu. Chaque relation doit contenir exactement trois éléments (id source, type de relation, id cible), et le type de relation doit être correctement spécifié. Si une relation ne respecte pas ces critères, elle est supprimée. Les relations valides sont ensuite utilisées pour identifier et conserver uniquement les entités pertinentes associées à cette relation cible.

Les entités retenues subissent des corrections pour assurer un format homogène, notamment la conversion des identifiants de chaîne de caractères en entiers. Les relations mal formatées ou non pertinentes sont éliminées, et les entités qui ne sont pas associées aux relations cibles sont également supprimées.

Ce tri extrêmement sélectif a entraîné une importante perte de données générées. En effet, près de 84 % des données ont été supprimées après ce filtrage. Le détail des résultats est présenté dans le tableau 4.

	Nombre de phrases après post-traitement	Nombre de phrases filtrées	Pourcentage de phrases filtrées
WAS_DISSOLVED_IN	311	9	3 %
WAS_CREATED_IN	418	86	21 %
CREATED	393	115	29 %
DIED_IN	579	53	9 %
IS_COOPERATING_WITH	615	45	7 %
IS_BORN_IN	894	198	22 %
Total	3 210	506	16 %

TAB. 4 – *Résultats après le filtrage sur les 6 catégories.*

4 Extraction de relations

4.1 Méthode proposée pour la classification de relations

Nous avons modélisé le problème d'extraction de relations sous la forme d'une classification de couples d'entités. Dans un premier temps, nous faisons une énumération exhaustive des couples d'entités possibles dans un document. En effet, lors de l'énumération des couples, ces derniers sont filtrés selon le guide des relations fourni dans le cadre du challenge. Seuls les couples pouvant avoir une relation sont gardés. Par exemple, un couple de type EVENT-PLACE peut avoir la relation *STARTED_IN*, il est donc retenu. Cependant, un couple de type PLACE-EVENT n'ayant pas de relation possible est donc rejeté. Dans un second temps les tokens spéciaux [e1] et [e2] sont insérés, respectivement, autour des mentions des entités 1 et 2. Le label de l'entité est également inséré autour de chaque mention. Cela donne, par exemple, "[e1][CIVILIAN]Anam Des-

tresse[CIVILIAN][e1]". Lors de l'entraînement, en entrée du modèle, par chaque couple d'entité pouvant avoir une relation, est envoyé le texte entier du document avec les informations insérées dans le texte comme présenté ci-dessus.

Etant donné les différences entre les relations entité-entité et les relations entité-attribut, nous avons fait le choix de mettre en place deux modèles.

Le premier modèle concerne les relations entité-attribut, où pour chaque couple d'entités, il y a une unique relation à prédire. Ainsi, la modélisation du problème est une classification en multi-classe.

Le deuxième modèle, pour les relations entité-entité, doit pouvoir prédire plusieurs relations pour un couple d'entités. Ainsi, la modélisation du problème est une classification en multi-classe et en multi-label.

En effet, nous avons testé plusieurs configurations, celle-ci obtient les meilleurs résultats. Les autres configurations sont :

— Différents niveaux de granularité pour les entités de type *ACTOR* ou *EVENT*. Le niveau de granularité le plus bas a été retenu.
— Différentes sections du texte. Seulement les phrases contenant au moins une mention du couple d'entité sont gardées. Le contexte des autres phrases est important. Pour les relations de type entité-attribut, nous avons remarqué que le texte contient au moins une phrase où se trouve au moins une mention de chaque entité du couple. Nous avons essayé de ne garder que cette phrase et avons observé des résultats dégradés.

4.2 Résultats et interprétations

Nous avons expérimenté deux architectures de transformeurs pré-appris : un BERT et mDeBERTav3 (He et al., 2021). Les résultats, sur le *dataset* de validation, des *fine-tuning* de mDeBERTa sur les deux groupes sont reportés dans le tableau 5 pour les relations entité-attribut et dans le tableau 7 pour les relations entité-entité. Le tableau 5 montre que mDeBERTa est le plus adapté à la tâche et sera le modèle utilisé pour les entraînements suivants.

Les relations entité-attribut sont des relations simples et sont bien reconnues par le modèle. Cependant plusieurs de ces relations ayant moins de 50 occurrences dans le *dataset* ont un F1-score de 1. Il est probable que le modèle ne soit pas robuste pour ces relations. Notamment pour les relations ayant moins de 20 occurrences qui possèdent entre 2 et 4 exemples dans le *dataset* de validation.

Les relations entité-entité sont plus complexes. Le nombre d'occurrence dans le *dataset* d'entraînement est un facteur de la qualité mais pas uniquement. En effet, les résultats sont très mauvais pour les relations de moins de 100 exemples. Cependant, un nombre d'exemple élevé ne garantit pas de bons résultats. Par exemple, la relation *HAS_CONTROL_OVER* est la deuxième relation la plus représentée avec plus de 4000 occurrences. Pourtant, nous obtenons un F1-score de 0,5.

Concernant l'entraînement avec les données augmentées, nous avons dû l'arrêter avant la fin par faute de temps. Les scores sur la validation sont montrés dans la colonne 2 du tableau 7. Cela explique que les F1-scores soient moins élevés, pour les entités non augmentées, que pour le modèle sans augmentation. Néanmoins, le F1-score des relations augmentées est meilleur avec augmentation des données.

Après la fin du challenge, nous avons relancé un entraînement avec les données augmentées. Bien que les résultats n'apparaissent pas dans les résultats du challenge, il nous semble important de pouvoir conclure sur l'augmentation de données avec un entraînement qui a abouti. Les résultats de cet entraînement se trouvent dans la colonne 3 du tableau 7.

Dans un premier temps, nous observons une augmentation du score F1 sur les relations augmentées entre les colonnes 1 et 3 du tableau 7. Pour les relations *IS_BORN_IN*, *IS_COOPERATING_WITH* et *CREATED*, les scores F1 sont doublés avec l'augmentation de données. Les relations *WAS_CREATED_IN* et *WAS_DISSOLVED_IN* sont maintenant détectées par le modèle. Cependant, la relation *DIED_IN* n'a pas d'augmentation significative du score. Les 53 exemples synthétiques ajoutés n'ont pas suffi à augmenter le score.

Dans un second temps, nous observons des effets de bord avec une baisse du score F1 pour les relations non augmentées ayant moins de 500 occurrences. Il semble que les données augmentées ajoutent un bruit qui dégrade le score des relations avec peu d'occurrences. La relation qui subit le plus cette dégradation est la relation *IS_BORN_ON*. Etant la relation avec le moins d'exemples, seulement 20, le score chute de 0.49 points. Cette baisse de score est due à une augmentation des faux positifs. De plus, on remarque que la relation *IS_BORN_IN* est la relation augmentée dont le score augmente le plus, passant de 0.19 à 0.48. Les deux relations portant sur la naissance d'un individu, il est possible que ces évolutions soient corrélées.

Notre soumission finale au challenge est la concaténation des prédictions de 3 modèles :

— Les relations entité-attribut sont prédites par le modèle dont les résultats sont dans le tableau 5.
— Les relations pour lesquelles nous avons ajouté des données sont prédites par le modèle dont les résultats sont dans la deuxième colonne du tableau 7. C'est le modèle entraîné sur les données augmentées.
— Le reste des relations sont prédites par le modèle dont les résultats sont dans la première colonne du tableau 7. C'est le modèle entraîné sans les données augmentées.

	bert	mdeberta	n train
DEATHS_NUMBER	0,23	**0,79**	75
HAS_CATEGORY	0,97	**0,99**	894
HAS_COLOR	0,86	**0,90**	91
HAS_FOR_HEIGHT	0,00	**1,00**	12
HAS_FOR_LENGTH	0,57	**1,00**	16
HAS_FOR_WIDTH	0,00	**1,00**	14
HAS_LATITUDE	0,00	**0,57**	10
HAS_LONGITUDE	0,00	**1,00**	12
HAS_QUANTITY	0,94	**0,95**	191
INJURED_NUMBER	0,00	**0,30**	70
IS_OF_NATIONALITY	0,84	**0,91**	179
IS_OF_SIZE	**0,96**	0,95	438
IS_REGISTERED_AS	0,75	**1,00**	34
WEIGHS	0,60	**1,00**	41
GENDER_FEMALE	0,90	**0,93**	414
GENDER_MALE	0,92	**0,95**	908

TAB. 5 – *F1 score par label pour les relations de type entité-attribut sur le dataset de validation.*

5 Optimisation des résultats : pistes d'amélioration et réflexions

5.1 Augmentation de données

Plusieurs axes d'amélioration peuvent être envisagés pour affiner davantage les résultats obtenus. Tout d'abord, il serait pertinent de relancer les expérimentations en modifiant les *prompts*, en explorant différentes formulations, ou en adoptant des modèles de langage plus puissants et récents comme *GPT-4* (OpenAI et al., 2023) ou *Mistral 7B* (Jiang et al., 2023) (voire même un modèle plus conséquent comme *Mistral large 123B*). Toutefois, il serait important d'évaluer le ratio coût/efficacité avant de privilégier l'utilisation de ces modèles plus complexes, souvent très gourmands en ressources matérielles et énergétiques. Cette évaluation permettrait de déterminer si les gains potentiels en termes de performance justifient les coûts additionnels.

Une autre stratégie consisterait à revoir la structuration des *prompts*. Par exemple, une approche pourrait impliquer la génération d'une phrase contenant les entités en premier, suivie uniquement des relations à annoter. Cette reformulation pourrait potentiellement améliorer la précision des sorties en clarifiant les attentes pour le modèle.

Enfin, une approche hybride mériterait d'être explorée : utiliser un modèle pour générer les phrases et un autre, spécialisé, pour effectuer l'annotation des relations. En combinant les forces spécifiques de différents modèles, cette stratégie pourrait augmen-

ter la robustesse des résultats tout en minimisant les biais spécifiques à un seul modèle.

Ces ajustements, pris individuellement ou de manière combinée, ouvrent la voie à des résultats plus précis et plus fiables.

Cependant, il ne s'agit pas uniquement d'améliorations algorithmiques. Nous pensons également que le post-traitement des données générées permettrait de couvrir une partie des pertes. À cet égard, le second filtrage des données générées pourrait être amélioré en identifiant précisément les raisons pour lesquelles certaines sont supprimées et en analysant ce qui les rend incohérentes. Ce travail, pourrait fournir des pistes pour affiner les *prompts* et ajuster les critères du filtrage, augmentant ainsi le nombre de phrases conservées tout en maintenant la cohérence et de pertinence des données. Ce rééquilibrage, permettrait non seulement de compenser certaines lacunes des modèles, mais également de garantir une meilleure contextualisation des résultats.

5.2 Représentation à base de graphe

Plusieurs relations entité-entité ont des propriétés particulières par rapport aux autres relations. Nous remarquons de la transitivité pour des relations, notamment la relation *IS_LOCATED_IN*. En effet pour deux entités *PLACE* : "place1" et "place2", si "place1" is_located_in "place2" et qu'une autre entitéA is_located_in "place1" alors entitéA is_located_in "place2".

Nous avons imaginé deux méthodes pour initialiser un graphe au niveau du document. La première est de reprendre la méthode d'énumération des couples d'entités possibles utilisée dans la première approche pour générer le *dataset*. Ensuite, l'idée était de modifier la tâche en classification des arcs du graphe.
La deuxième méthode est de construire un graphe à partir des arbres de dépendance syntaxique pour chaque phrase. Le *dataset* étant annoté en coréférence, nous pouvons utiliser les mentions des entités pour combiner les graphes de chaque phrase en un seul graphe à l'échelle du document.
Nous avons utilisé SpaCy [4] pour extraire les arbres de dépendance des phrases. Exemple de graphe pour un document figure 2. Un tel graphe pourrait amener le contexte nécessaire à la détection des relations les plus complexes.

4. `https://spacy.io/usage/linguistic-features/#dependency-parse`

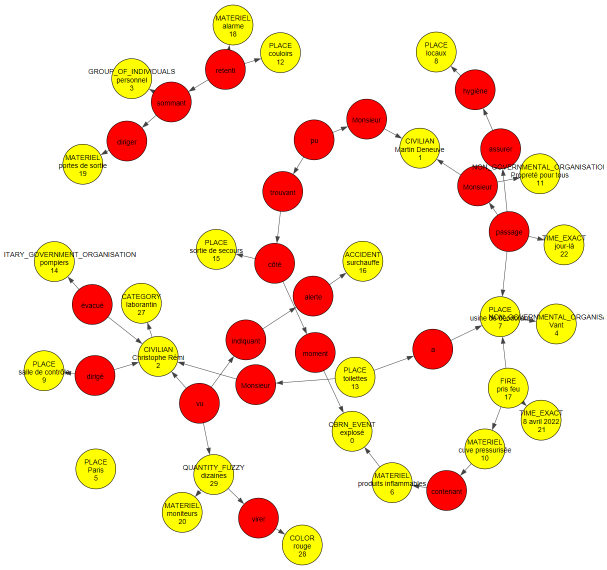


FIG. 2 – *Graphe des entités à partir des dépendances syntaxiques sur un document.*

5.3 Modèle de classification de relation

Différentes améliorations sont possibles concernant les modèles et leurs sorties. Tout d'abord, pour le modèle entité-attribut, faire une augmentation de données simple pour les relations avec moins de 20 occurrences pour s'assurer de la robustesse du modèle. Les relations étant simples une augmentation à base de dictionnaire ou de reformulation pourrait être suffisante.

Ensuite, pour le modèle entité-entité, faisant de la classification multi-classe multi-label, un seuil a été défini empiriquement à partir duquel on accepte la prédiction du modèle. Ce seuil est le même pour toutes les entités. Un seuil par classe pourrait être plus pertinent.

La représentation par graphe semble être pertinente par rapport à la structure des données. Soit par la modélisation du problème en fournissant le graphe en entrée du modèle. Soit en faisant des post-traitements, notamment en complétant le graphe des relations prédites par un modèle. Un tel travail pourrait permettre de trouver les relations transitives implicites, observé par exemple pour la relation *IS_LOCATED_IN*.

Enfin, les modèles joints en reconnaissance d'entités et en extraction de relation ont montré leur performance dans la littérature. Une exploration de ces approches semble être prometteuse. Dans le cadre de ce challenge, nous avons initié des expérimentations, non concluantes, dans ce sens. En effet, nous avons expérimenté le modèle EnRiCO (Zaratiana et al., 2024), basé sur un mécanisme d'attention simultané sur les

entités et les relations, dans un contexte one-shot. Comme ce modèle est pré-entrainé en anglais, nous avons opté par une traduction automatique des données du challenge vers l'anglais en pré-traitement. Une analyse qualitative montre des résultats décevants. Un fine-tunning du modèle EnRiCO sur les données du challenge nous semble une piste pertinente.

6 Conclusion

L'objectif de cette étude était d'explorer les méthodes d'extraction de relations entre entités textuelles en utilisant un corpus augmenté grâce à l'application de modèles de langage de grande taille (LLM). Nos expérimentations ont révélé que l'un des principaux défis réside dans le déséquilibre des relations dans le corpus, ce qui impacte particulièrement les relations rares. Pour surmonter cette difficulté, nous avons utilisé des modèles LLM pour générer des données supplémentaires, ce qui a permis d'améliorer en partie la couverture de certaines catégories de relations, en particulier celles initialement sous-représentées en micro-F1. Par exemple, les relations comme *IS_BORN_IN* et *IS_COOPERATING_WITH* qui avaient un faible F1 sans augmentation, ont montré des améliorations grâce à l'augmentation des données.

Pendant le challenge, nous avons observé que l'augmentation des données permet une légère amélioration des résultats, notamment pour les relations complexes ou peu fréquentes. En effet, parmi les 6 classes pour lesquelles l'augmentation des données a été effectuée, seules 5 classes montrent un gain en performance (par exemple, la relation *IS_BORN_IN*, dont le F1 score passe de 0,19 à 0,29 avec augmentation de données). Cependant, ces résultats sont issus d'un entraînement qui n'a pas été mené à son terme. Ainsi, à la fin du challenge, nous avons effectué un nouvel entraînement complet avec les données augmentées. Bien que ces résultats ne soient pas inclus dans les résultats finaux du challenge, nous avons observé une amélioration notable des scores F1 sur les relations augmentées, avec des scores doublés grâce à l'augmentation des données (par exemple, la relation *IS_BORN_IN*, dont le F1 score passe de 0,19 à 0,48). Toutefois, un peu de bruit persiste sur les relations non augmentées, notamment celles avec peu d'occurrences.

Des problèmes persistent, notamment en ce qui concerne le format des données générées et l'exactitude des informations, telles que la position des entités dans les phrases. Un post-traitement des sorties du modèle a été nécessaire pour nettoyer et garantir leur conformité aux exigences d'annotation. Malgré ces efforts, des pertes de données significatives ont été constatées, dues en grande partie à la redondance ou aux erreurs dans la génération des données.

Nos résultats montrent que l'optimisation des *prompts* et l'expérimentation avec des modèles encore plus puissants, comme GPT-4, pourraient améliorer la précision et la fiabilité des phrases générées. Une approche hybride, combinant génération et annotation par des modèles spécialisés, pourrait également s'avérer bénéfique. De plus, les comparaisons entre les modèles mDeBERTa et BERT ont révélé que mDeBERTa

est mieux adapté à la tâche d'extraction de relations entité-attribut, comme l'illustre le tableau 5, où les scores F1 pour mDeBERTa sont supérieurs à ceux de BERT dans notre contexte.

Par ailleurs, des expérimentations supplémentaires ont été menées avec des graphes, qui semblaient prometteuses pour l'extraction de relations, mais elles n'ont pas pu être approfondies en raison de contraintes de temps. Ces approches apparaissent néanmoins pertinentes, notamment par rapport à la structure des données, et mériteraient d'être explorées davantage dans de futures recherches.

En conclusion, bien que des améliorations aient été apportées grâce à l'augmentation des données et à l'utilisation de modèles puissants comme mDeBERTa, des ajustements supplémentaires sont nécessaires pour surmonter les défis liés à la génération de données et à leur traitement afin d'obtenir des résultats encore plus robustes et fiables. L'intégration de données augmentées a d'ores et déjà montré un impact positif sur les performances des modèles, mais une optimisation continue est nécessaire pour maximiser l'efficacité de l'extraction des relations complexes.

Références

Armingaud, R., A. Peuvot, R. Besançon, O. Ferret, S. Souihi, et al. (2024). Cea-list@evalllm2024 : prompter un très grand modèle de langue ou affiner un plus petit ? In *Atelier sur l'évaluation des modèles génératifs (LLM) et challenge d'extraction d'information few-shot*, Toulouse, France. Institut des sciences informatiques et de leurs interactions - CNRS Sciences informatiques [INS2I-CNRS].

Bassignana, E. et B. Plank (2022). What do you mean by relation extraction ? a survey on datasets and study on scientific relation classification. In S. Louvan, A. Madotto, et B. Madureira (Eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics : Student Research Workshop*, Dublin, Ireland, pp. 67–83. Association for Computational Linguistics.

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems 33*, 1877–1901.

Delaunay, J., H. T. H. Tran, C.-E. González-Gallardo, G. Bordea, N. Sidere, et A. Doucet (2023). A comprehensive survey of document-level relation extraction (2016-2023).

Devlin, J., M.-W. Chang, K. Lee, et K. Toutanova (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, et T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics.

He, P., J. Gao, et W. Chen (2021). Debertav3 : Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *CoRR abs/2111.09543*.

Jiang, A. Q., A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, et W. E. Sayed (2023). Mistral 7b.

OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, et B. Zoph (2023). Gpt-4 technical report.

Zaratiana, U., N. Tomeh, Y. Dauxais, P. Holat, et T. Charnois (2024). Enrico : Enriched representation and globally constrained inference for entity and relation extraction.

Annexe 1

```
Génère un paragraphe annoté de 100 mots maximum, au format JSON, sans
introduction ni explications, décrivant la date de naissance d'une personne
civile.

Par exemple : {
    'id': '186',
    'text': 'À Bruxelles, Annette Deneuve a perdu ses bras sur le chantier
de construction d'une voie ferrée à Louise. Un coffrage en béton de
plusieurs tonnes est tombé de la flèche en porte-à-faux d'une grue alors
qu'elle travaillait courbée sur les rails. Suite à cet accident qui s'est
produit le 4 octobre 2017, elle a été amputée des deux bras. Cette femme de
nationalité américaine est née le 20 mars 1995 en Pennsylvanie et réside à
Bruxelles pour son travail. D'après l'Association des cheminots, ces morts
tragiques sont devenues répétitives et ont plongé les travailleurs dans une
grande insécurité.',
    'entities': [
        {'id': 2, 'mentions': [
            {'value': 'Annette Deneuve', 'start': 13, 'end': 28},
            {'value': 'elle', 'start': 206, 'end': 210},
            {'value': 'elle', 'start': 304, 'end': 308},
            {'value': 'femme', 'start': 344, 'end': 349}
            ], 'type': 'CIVILIAN'
        },
        {'id': 6, 'mentions': [
            {'value': 'Pennsylvanie', 'start': 403, 'end': 415}
        ], 'type': 'PLACE'}
    ],
    'relations': [[2, 'IS_BORN_IN', 6]]
}.

Dans cet exemple : 'id' : un identifiant unique pour chaque phrase générée.
'entities' : une liste d'entités mentionnées dans la phrase, chacune avec
son propre identifiant et ses mentions. Si une entité est mentionnée
plusieurs fois, ses mentions partagent le même identifiant. 'start' et
'end' indiquent les positions des caractères de début et de fin de chaque
mention. 'relations' : décrit la relation entre deux entités par leur
identifiant, par exemple [[2, 'IS_BORN_IN', 6]] indique que l'entité 2 a
initié une interaction avec l'entité 6.
```

FIG. 3 – *Exemple de prompt utilisée pour l'augmentation de données de la relation BORN_IN.*

Annexe 2

	unique	multiple
STARTED_IN	213	1647
IS_LOCATED_IN	6460	2565
OPERATES_IN	2256	179
IS_PART_OF	989	473
CREATED	25	101
HAS_CONTROL_OVER	3018	1529
IS_IN_CONTACT_WITH	2001	918
IS_COOPERATING_WITH	154	218
INITIATED	468	**1**
HAS_CONSEQUENCE	769	**0**
IS_AT_ODDS_WITH	1107	419
RESIDES_IN	19	203
IS_BORN_IN	22	28
HAS_FAMILY_RELATIONSHIP	123	99
DIED_IN	40	**1**
DEATHS_NUMBER	73	**2**
HAS_CATEGORY	894	**0**
HAS_COLOR	91	**0**
HAS_FOR_HEIGHT	12	**0**
HAS_FOR_LENGTH	16	**0**
HAS_FOR_WIDTH	14	**0**
HAS_LATITUDE	10	**0**
HAS_LONGITUDE	10	**0**
HAS_QUANTITY	191	**0**
INJURED_NUMBER	69	**1**
IS_OF_NATIONALITY	179	**0**
IS_OF_SIZE	437	**1**
IS_REGISTERED_AS	34	**0**
WEIGHS	41	**0**
GENDER_FEMALE	413	**1**
GENDER_MALE	906	**2**
IS_BORN_ON	20	**0**
IS_DEAD_ON	68	**0**
WAS_CREATED_IN	15	**0**
WAS_DISSOLVED_IN	14	**0**
START_DATE	233	801
END_DATE	73	801

TAB. 6 – *Nombre de relations unique ou multiple pour chaque couple d'entité, par le label de la relation.*

Annexe 3

	sans augmentation	avec augmentation *1	avec augmentation *2	nombre d'occurences (+ augmentation)
HAS_FAMILY_RELATIONSHIP	**0,91**	0,82	0,82	222
START_DATE	**0,83**	0,78	0,82	1034
OPERATES_IN	0,73	0,74	**0,76**	2435
END_DATE	0,77	0,70	**0,79**	874
STARTED_IN	0,69	0,69	**0,70**	1860
IS_LOCATED_IN	0,69	0,68	**0,73**	9025
HAS_CONSEQUENCE	**0,52**	0,50	**0,52**	769
HAS_CONTROL_OVER	0,50	0,50	**0,52**	4547
IS_AT_ODDS_WITH	**0,50**	0,47	0,49	1526
IS_BORN_ON	**0,80**	0,44	0,31	20
IS_IN_CONTACT_WITH	**0,48**	0,44	0,47	2919
IS_PART_OF	0,45	0,42	**0,46**	1462
INITIATED	**0,45**	0,41	0,40	469
IS_DEAD_ON	**0,50**	0,38	0,36	68
IS_BORN_IN	0,19	0,29	**0,48**	50 (+ 198)
RESIDES_IN	**0,41**	0,24	0,34	222
IS_COOPERATING_WITH	0,11	0,19	**0,20**	372 (+ 45)
DIED_IN	0,09	**0,15**	0,11	41 (+ 53)
CREATED	0,06	0,11	**0,13**	126 (+ 115)
WAS_CREATED_IN	0	0	**0,17**	15 (+ 86)
WAS_DISSOLVED_IN	0	0	**0,10**	14 (+ 9)

TAB. 7 – *F1 score par label avec et sans augmentation de données pour les relations entité-entité sur le dataset de validation. *1 sont les résultats de notre soumission finale, dont l'entraînement a été arrêté avant la fin. Cet entraînement à été refait après la fin du challenge, les résultats sont dans la colonne *2.*

Participation de l'équipe Défense au défi TextMine'25

Summary

Information extraction, in particular the extraction of relations in a specific context, remains a challenge despite the progress made in the field of Natural Language Processing (NLP). This article is part of the TextMine'25 challenge, proposed at the EGC 2025 conference, which focuses on the extraction of relationships between entities and events of military interest in simulated intelligence reports. We present an in-depth analysis of the corpus used, as well as a data augmentation method based on Large Language Models. This approach aims to enrich annotations and diversify the examples used for model training.

Extraction de relations multi-étiquettes en utilisant des modèles pré-entraînés et des couches de Transformer

Ngoc Luyen Le [*,**] Gildas Tagny Ngompé

[*]Gamaizer.ia, 93340 Le Raincy, France

[**]Université de technologie de Compiègne, CNRS, Heudiasyc

(Heuristics and Diagnosis of Complex Systems), CS 60319 - 60203 Compiègne Cedex, France

Résumé. Nous présentons dans cet article le modèle **BTransformer18**, une architecture d'apprentissage profond conçue pour l'extraction de relations multi-étiquettes dans des textes en français. Notre approche combine les capacités de représentation contextuelle de modèles de langage pré-entraînés de la famille BERT, tels que BERT, RoBERTa, ainsi que leurs versions francophones CamemBERT et FlauBERT, avec la puissance des encodeurs Transformer pour capturer les dépendances à long terme entre les tokens. Les expérimentations menées sur le jeu de données du défi TextMine'25 montrent que notre modèle atteint des performances supérieures, en particulier en utilisant **CamemBERT-Large**, avec un score F1-macro de 0,654, surpassant les résultats obtenus avec **FlauBERT-Large**. Ces résultats démontrent l'efficacité de notre approche pour l'extraction automatique de relations complexes dans des rapports de renseignement.

1 Introduction

L'extraction de relations est une tâche fondamentale en traitement automatique du langage naturel (TALN), visant à identifier et classifier les relations sémantiques entre des entités nommées au sein d'un texte (Nasar et al., 2021). Dans le cas des rapports de renseignement, cette tâche revêt une importance particulière pour la structuration de l'information et la facilitation de l'analyse. Les approches traditionnelles reposent souvent sur des méthodes d'apprentissage supervisé nécessitant des ressources annotées considérables et peinent à capturer les relations complexes présentes dans les textes non structurés.

Le défi TextMine'25 (Prieur et al., 2024) fournit un jeu de données précieux pour faire progresser la recherche dans ce domaine. Ce jeu de données est composé de 800 rapports de renseignement factices en français, annotés avec des mentions d'entités, leurs types, attributs et relations. Il pose un défi significatif en raison de la complexité des relations et de la nécessité de modèles capables de gérer la classification multi-étiquettes.

Avec l'émergence des modèles de langage pré-entraînés, tels que ceux de la famille BERT, des améliorations significatives ont été réalisées dans diverses tâches de TALN. BERT (Devlin, 2018) et RoBERTa (Liu, 2019) ont établi de nouveaux standards pour la langue anglaise, tandis

que leurs équivalents francophones, CamemBERT (Martin et al., 2019) et FlauBERT (Le et al., 2019), ont été développés pour traiter les spécificités de la langue française. Ces modèles, membres de la famille BERT, exploitent un pré-entraînement à grande échelle sur des corpus massifs pour produire des embeddings contextualisés riches.

Parallèlement, les encodeurs Transformer (Vaswani, 2017; Wolf et al., 2020) ont démontré une capacité exceptionnelle à modéliser les dépendances à long terme dans les séquences grâce aux mécanismes d'attention. Combiner les forces des modèles de langage pré-entraînés et des encodeurs Transformer offre une direction prometteuse pour améliorer les capacités d'extraction de relations.

Dans cet article, nous introduisons **BTransformer18**, un modèle qui intègre les modèles de langage pré-entraînés de la famille BERT avec des encodeurs Transformer pour l'extraction de relations multi-étiquettes dans des textes en français. Notre architecture comprend trois couches principales : **Embeddings Contextuels**, où nous obtenons les embeddings des tokens en utilisant des modèles comme CamemBERT et FlauBERT ; **Encodeurs Transformer**, qui capturent les dépendances complexes entre les tokens ; et **Agrégation et Classification**, où nous agrégeons les représentations et prédisons les relations.

Nous évaluons notre modèle sur le jeu de données du défi TextMine'25, en réalisant des expérimentations approfondies pour évaluer l'impact des différents modèles pré-entraînés sur la tâche. Nos résultats montrent que l'utilisation de **CamemBERT-Large** conduit à des gains de performance significatifs, atteignant un score F1-macro de 0,654, surpassant les modèles basés sur **FlauBERT-Large**.

Dans la section 2, nous détaillons notre approche proposée en décrivant l'architecture de **BTransformer18**. La section 3 présente le cadre expérimental, y compris le jeu de données, les étapes de prétraitement, les procédures d'entraînement, et des résultats de notre modèle. Enfin, dans la section 4, nous concluons et proposons des perspectives pour de futures recherches.

2 Approche proposée

Notre approche, illustrée dans la figure 1, s'appuie sur la logique de *fine-tuning* d'un modèle de langage pré-entraîné, composée d'un *body* (généraliste pour la langue ou le domaine) et d'une *head* (spécifique à la tâche d'extraction de relations). Dans notre modèle, le *body* est assuré par **CamemBERT-Large**, choisi pour sa spécialisation en français. La *head* de classification inclut une couche cachée basée sur l'architecture Transformer (Vaswani, 2017), permettant d'exploiter le mécanisme d'attention pour l'extraction de relations. La sortie est ensuite produite par une simple couche dense, qui assure la classification multi-label.

2.1 Embeddings contextuels

Un des modèles de langage pré-entraînés de la famille BERT (par exemple, BERT, CamemBERT, FlauBERT, etc.) est employé afin d'obtenir les embeddings contextuels des tokens du texte d'entrée. Soit $\mathbf{X} = [x_1, x_2, \ldots, x_T]$ la séquence de tokens, où T est la longueur de la séquence. Ces modèles de langage pré-entraînés génère pour chaque token x_t un embedding contextuel $\mathbf{h}_t \in \mathbb{R}^d$, où d est la dimension cachée du modèle, c'est-à-dire la taille du vecteur latent (par exemple, 768 pour la version BERT-base et 1024 pour la version BERT-Large).

$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T] = \text{BERT}(\mathbf{X})$$

Les embeddings contextuels $\mathbf{H}$ désignent les représentations vectorielles produites par un modèle de langage pré-entraîné pour chaque *token* du texte d'entrée. Contrairement aux *word embeddings* classiques (tels que `word2vec` (Mikolov, 2013) ou `GloVe` (Pennington et al., 2014)), qui associent à chaque mot un unique vecteur indépendant du contexte, les embeddings contextuels tiennent compte du voisinage lexical et syntaxique. Ainsi, le même mot peut avoir des représentations différentes selon son sens et sa position dans la phrase.

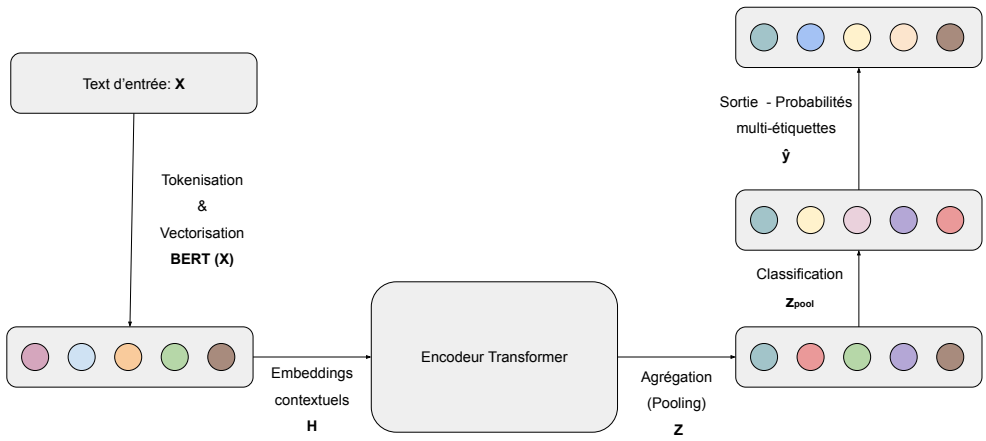


FIG. 1 – *Représentation schématique de l'architecture* ***BTransformer18****.*

2.2 Encodeurs transformer

Les embeddings contextuels $\mathbf{H}$ sont ensuite transmis à travers L couches d'encodeurs Transformer (Vaswani, 2017), afin de capturer les dépendances à long terme entre les tokens. Cette architecture permet de modéliser efficacement la structure du texte et de faire émerger des représentations de plus en plus riches à mesure que les couches s'empilent. Chaque couche ℓ transforme la représentation $\mathbf{Z}^{(\ell-1)}$ en $\mathbf{Z}^{(\ell)}$ comme suit :

$$\mathbf{Z}^{(\ell)} = \text{TransformerEncoder}^{(\ell)}\left(\mathbf{Z}^{(\ell-1)}\right), \quad \ell = 1, \ldots, L, \tag{1}$$

où $\mathbf{Z}^{(0)} = \mathbf{H}$. Le fonctionnement de chaque couche peut être décomposé en quatre étapes principales : Attention multi-têtes, 1er Add & Norm, Feed-forward positionnel et 2ème Add & Norm.

Attention multi-têtes : La première étape consiste à appliquer un mécanisme d'attention multi-têtes, noté MultiHeadAttention, sur $\mathbf{Z}^{(\ell-1)}$:

$$\mathbf{A}^{(\ell)} = \text{MultiHeadAttention}\left(\mathbf{Z}^{(\ell-1)}\right) \tag{2}$$

Ce module scinde l'espace latent en plusieurs *têtes* d'attention, permettant au modèle de se focaliser sur différents aspects du contexte. L'attention proprement dite est calculée comme suit :

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\Big)\mathbf{V}, \tag{3}$$

où $\mathbf{Q}$, $\mathbf{K}$ et $\mathbf{V}$ sont respectivement les matrices des requêtes, des clés et des valeurs, et d_k la dimension des clés. Grâce à ce mécanisme, le modèle peut se focaliser sur diverses parties de la séquence, offrant ainsi une capture fine des dépendances sémantiques et syntaxiques.

1er Add & Norm : Une fois la matrice d'attention $\mathbf{A}^{(\ell)}$ calculée, on l'additionne à l'entrée de la couche $\mathbf{Z}^{(\ell-1)}$. Cette somme est ensuite normalisée :

$$\mathbf{U}^{(\ell)} = \text{LayerNorm}\Big(\mathbf{Z}^{(\ell-1)} + \mathbf{A}^{(\ell)}\Big) \tag{4}$$

L'addition permet de conserver les informations initiales (via le *skip connection*), tandis que la normalisation de couche (`LayerNorm`) (Ba, 2016) stabilise l'apprentissage et facilite la convergence.

Feed-forward positionnel : Chaque position dans $\mathbf{U}^{(\ell)}$ est ensuite traitée de manière indépendante par un réseau pleinement connecté, souvent appelé *position-wise feed-forward network* (FFN). On calcule :

$$\mathbf{F}^{(\ell)} = \text{FFN}\Big(\mathbf{U}^{(\ell)}\Big) \tag{5}$$

Ce réseau permet d'enrichir localement la représentation de chaque token, en transformant linéairement puis en appliquant une fonction d'activation non linéaire (par exemple, ReLU).

2ème Add & Norm : Enfin, on ajoute la sortie du *feed-forward* $\mathbf{F}^{(\ell)}$ à $\mathbf{U}^{(\ell)}$, puis on normalise à nouveau :

$$\mathbf{Z}^{(\ell)} = \text{LayerNorm}\Big(\mathbf{U}^{(\ell)} + \mathbf{F}^{(\ell)}\Big) \tag{6}$$

À l'issue de cette étape, on obtient la sortie finale de la ℓ-ième couche, qui sert d'entrée à la couche suivante. En répétant ces opérations L fois, l'encodeur Transformer parvient à agréger des informations à différentes échelles, améliorant ainsi sa capacité à modéliser les relations sémantiques et structurelles présentes dans la séquence d'entrée.

2.3 Agrégation et classification

La sortie du dernier encodeur Transformer est une séquence de représentations $\mathbf{Z}^{(L)} = \big[\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T\big]$, où $\mathbf{z}_t \in \mathbb{R}^d$ représente la représentation contextuelle du t-ième token. Pour réduire cette séquence à un unique vecteur reflétant l'information globale, nous effectuons une agrégation par moyenne :

$$\mathbf{z}_{\text{pool}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{z}_t \tag{7}$$

Cette opération, appelée *mean pooling*, permet de combiner les informations contenues par chaque token, tout en préservant la structure sémantique globale de la séquence. Elle est souvent moins sensible au bruit que d'autres méthodes d'agrégation (telles que l'utilisation d'un jeton spécial `[CLS]`) et tend à mieux lisser les variations locales.

Une fois la représentation globale $\mathbf{z}_{\text{pool}}$ obtenue, nous employons une couche de classification pour prédire les relations associées à la séquence. Plus précisément, nous appliquons une transformation linéaire, puis une fonction sigmoïde :

$$\hat{\mathbf{y}} = \sigma\Big(\mathbf{W}\,\mathbf{z}_{\text{pool}} + \mathbf{b}\Big), \tag{8}$$

où $\mathbf{W} \in \mathbb{R}^{C\times d}$ et $\mathbf{b} \in \mathbb{R}^{C}$. Ici, C désigne le nombre de classes de relations, et σ est la fonction sigmoïde appliquée à chaque composante. La sortie $\hat{\mathbf{y}} \in [0,1]^C$ est alors interprétée comme un vecteur de probabilités indiquant la présence ou l'absence de chaque relation.

3 Expérimentations et résultats

Dans cette section, nous décrivons les expérimentations menées pour évaluer les performances de notre modèle **BTransformer18** sur la tâche d'extraction de relations multi-étiquettes. Nous présentons d'abord le jeu de données utilisé, puis les détails de l'implémentation et des paramètres d'entraînement. Enfin, nous discutons des résultats obtenus.

3.1 Jeu de données et prétraitement des données

Le jeu de données utilisé pour les expérimentations est constitué de 800 rapports de renseignement factices fournis dans le cadre du défi TextMine'25 (Prieur et al., 2024). En général, les données d'entrée se composent d'un texte, des mentions et des types des entités ainsi que des attributs associés.

Par rapport à la tokenisation : les textes sont tokenisés en utilisant le tokenizer associé au modèle de language pré-entraîné, garantissant une correspondance optimale avec les embeddings. Les entités annotées sont alignées avec les tokens pour former des paires d'entités potentielles (e_i, e_j).

Sur la construction des Paires d'Entités : pour chaque document, nous générons toutes les paires possibles d'entités annotées, où e_i et e_j sont des entités du texte. Chaque paire est associée à un vecteur de labels multi-étiquettes $\mathbf{y}_{ij} \in \{0,1\}^C$, indiquant les relations existantes entre e_i et e_j.

3.2 Entraînement du modèle

Fonction de perte : La Binary Cross-Entropy (BCE) est utilisée comme fonction de perte pour la classification multi-étiquettes. Cette fonction est définie par l'équation suivante :

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}\left[y_{nc}\log(\hat{y}_{nc}) + (1-y_{nc})\log(1-\hat{y}_{nc})\right],$$

où N représente le nombre total de paires d'entités dans le lot d'entraînement, $y_{nc} \in \{0, 1\}$ est l'étiquette réelle pour la classe c, et $\hat{y}_{nc} \in [0, 1]$ est la probabilité prédite par le modèle pour cette classe.

Optimisation : L'optimisation du modèle est réalisée à l'aide de l'optimiseur **AdamW** (Loshchilov, 2017), avec un taux d'apprentissage initial $\alpha_0 = 2 \times 10^{-5}$. Un scheduler avec *warm-up* est appliqué pour ajuster dynamiquement le taux d'apprentissage au cours de l'entraînement. La mise à jour du taux d'apprentissage est donnée par :

$$\alpha_t = \alpha_0 \times \min\left(\frac{t}{t_{\text{warmup}}}, 1\right),$$

où t est le nombre d'itérations effectuées, et t_{warmup} est le nombre d'itérations pendant la phase de warm-up.

Régularisation : Pour réduire le risque de surapprentissage, plusieurs techniques de régularisation sont intégrées dans l'architecture. Un **dropout** (Srivastava et al., 2014) avec un taux de $p = 0.1$ est appliqué dans les couches Transformer et la couche de classification. En parallèle, une régularisation **L2** (pondération de décroissance) est utilisée sur les poids du modèle pour limiter leur amplitude et améliorer la généralisation.

Évaluation : Les performances du modèle sont mesurées à l'aide de métriques spécifiques à la classification multi-étiquettes. En particulier, la F1-mesure macro est employée pour évaluer la précision et le rappel moyens sur toutes les classes, pondérant chaque classe de manière égale (Prieur et al., 2024).

3.3 Résultats

Les résultats obtenus par notre modèle sont présentés dans le tableau 2, avec une comparaison entre deux modèles de langage pré-entraînés pour le français. Les expérimentations ont été réalisées en utilisant le modèle **BTransformer18**, en suivant les configurations et les hyperparamètres décrits dans la table 3.2.

Les résultats montrent que le modèle **BTransformer18** utilisant **CamemBERT-Large** obtient un score de **0,654**, tandis que celui utilisant **FlauBERT-Large** atteint un score de **0,620**, indiquant une amélioration de **3,4 points de pourcentage** grâce à l'intégration de **CamemBERT-Large**. Comme illustré dans la figure 2, les courbes d'entraînement de **BTransformer18** avec **CamemBERT** montrent une convergence rapide des pertes d'entraînement et de validation au cours des premières époques, suivie d'une stabilisation. Simultanément, la F1-mesure macro de validation progresse rapidement avant d'atteindre un plateau, indiquant une amélioration significative des performances de classification multi-étiquettes. L'écart modéré entre les pertes d'entraînement et de validation met en évidence une bonne généralisation du modèle, sans signe de surapprentissage, même après plusieurs époques. Ces résultats soulignent l'efficacité de **BTransformer18** à exploiter les représentations contextuelles riches de **CamemBERT-Large**, permettant d'apprendre des relations complexes tout en maintenant des performances stables sur les données de validation.

Hyperparamètre	**Valeur**
Modèle de base	CamemBERT-Large
Taille des embeddings	1024
Nombre de couches Transformer	2
Nombre de têtes d'attention	8
Taux de dropout	0.1
Longueur maximale de séquence	150
Taille du batch (entraînement)	16
Taille du batch (validation)	16
Taux d'apprentissage initial (α_0)	2×10^{-5}
Optimiseur	AdamW
Scheduler	Linear avec warm-up
Proportion du warm-up	10%
Nombre maximal d'époques	50
Patience pour l'arrêt anticipé	3
Classes de classification	37 (multi-étiquettes)

TAB. 1 – *Hyperparamètres utilisés pour l'entraînement du modèle **BTransformer18**.*

Les résultats suggèrent que le choix du modèle de langage pré-entraîné a un impact significatif sur les performances de l'extraction de relations multi-étiquettes. L'utilisation de **CamemBERT-Large** semble offrir une meilleure représentation contextuelle pour les textes du défi, ce qui se traduit par une amélioration notable des performances.

Pour garantir la reproductibilité des résultats et permettre une exploration plus approfondie, le code source complet de l'implémentation est disponible publiquement sur le dépôt GitHub suivant : `https://github.com/lengocluyen/relation_extraction_textmine25`.

4 Conclusion et perspectives

Dans cet article, nous avons présenté le modèle **BTransformer18**, une architecture combinant des modèles de langage pré-entraînés francophones, tels que **CamemBERT-Large** et **FlauBERT-Large**, avec des couches Transformer pour l'extraction de relations multi-étiquettes dans des rapports de renseignement. Les résultats expérimentaux ont démontré la supériorité de **CamemBERT-Large**, qui a obtenu un score F1 macro supérieur à celui de **FlauBERT-Large**. L'analyse des courbes d'entraînement a mis en évidence une convergence rapide et une bonne

Modèle	**Score**
BTransformer18 (CamemBERT-Large)	0,654
BTransformer18 (FlauBERT-Large)	0,620

TAB. 2 – *Résultats des modèles sur le jeu de données du défi TextMine'25.*

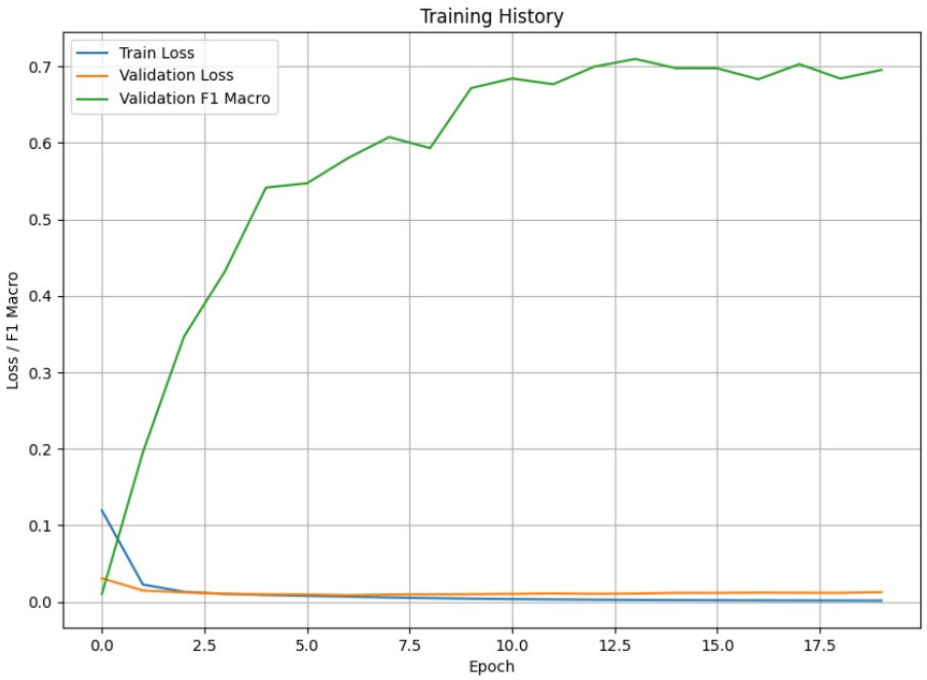


FIG. 2 – *Évolution des pertes et de la F1-mesure macro de validation pour le modèle* ***BTransformer18*** *avec* ***CamemBERT-Large****.*

généralisation, montrant l'efficacité du modèle pour capturer des relations complexes tout en évitant le surapprentissage. En exploitant les avancées récentes en traitement du langage naturel, notre modèle démontre sa capacité à relever le défi de l'extraction automatique de relations complexes, tout en maintenant une classification précise grâce aux couches Transformer.

Bien que les résultats obtenus soient prometteurs, plusieurs axes d'amélioration peuvent être explorés dans de futurs travaux. Tout d'abord, l'enrichissement des données avec des corpus supplémentaires ou annotés dans d'autres domaines pourrait renforcer la robustesse du modèle et améliorer sa généralisation. Par ailleurs, l'incorporation de graphes de connaissances ou l'utilisation de modèles d'apprentissage par graphes pourrait améliorer la modélisation des relations complexes entre entités. De plus, l'intégration de grands modèles de langage (*Large*

Language Models, LLMs), comme GPT (Achiam et al., 2023), Mistral (Jiang et al., 2023), LLama (Touvron et al., 2023), et des autres, pourrait offrir des représentations contextuelles encore plus riches et dynamiques, notamment pour des relations subtiles ou rares. Enfin, des techniques d'augmentation des données, combinées à des méthodes de régularisation avancées, ainsi qu'une optimisation des ressources computationnelles, permettraient de développer des modèles plus robustes et efficaces, adaptés à des applications en temps réel ou dans des environnements à ressources limitées. Ces pistes visent à élargir l'applicabilité du modèle tout en améliorant ses performances dans des tâches d'extraction de relations complexes.

Références

Achiam, J., S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv :2303.08774*.

Ba, J. L. (2016). Layer normalization. *arXiv preprint arXiv :1607.06450*.

Devlin, J. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

Jiang, A. Q., A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. (2023). Mistral 7b. *arXiv preprint arXiv :2310.06825*.

Le, H., L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, et D. Schwab (2019). Flaubert : Unsupervised language model pre-training for french. *arXiv preprint arXiv :1912.05372*.

Liu, Y. (2019). Roberta : A robustly optimized bert pretraining approach. *arXiv preprint arXiv :1907.11692 364*.

Loshchilov, I. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv :1711.05101*.

Martin, L., B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de La Clergerie, D. Seddah, et B. Sagot (2019). Camembert : a tasty french language model. *arXiv preprint arXiv :1911.03894*.

Mikolov, T. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781 3781*.

Nasar, Z., S. W. Jaffry, et M. K. Malik (2021). Named entity recognition and relation extraction : State-of-the-art. *ACM Computing Surveys (CSUR) 54*(1), 1–39.

Pennington, J., R. Socher, et C. D. Manning (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Prieur, A., G. Gadek, A. Guille, H. Rawsthorne, P. Cuxac, et C. Lopez (2024). Défi textmine'25 – extraction de relations pour analyser des rapports de renseignement. *Actes de l'atelier TextMine'25, Extraction et Gestion des Connaissances 2025 (EGC'25)*.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, et R. Salakhutdinov (2014). Dropout : a simple way to prevent neural networks from overfitting. *The journal of machine learning*

research 15(1), 1929–1958.

Touvron, H., T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. (2023). Llama : Open and efficient foundation language models. *arXiv preprint arXiv :2302.13971*.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. (2020). Transformers : State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing : system demonstrations*, pp. 38–45.

Summary

In this article, we present the BTransformer18 model, a deep learning architecture designed for multi-label relation extraction in French texts. Our approach combines the contextual representation capabilities of pre-trained language models from the BERT family—such as BERT, RoBERTa, and their French counterparts CamemBERT and FlauBERT—with the power of Transformer encoders to capture long-term dependencies between tokens. Experiments conducted on the dataset from the TextMine'25 challenge show that our model achieves superior performance, particularly when using CamemBERT-Large, with a macro F1 score of 0.654, surpassing the results obtained with FlauBERT-Large. These results demonstrate the effectiveness of our approach for the automatic extraction of complex relations in intelligence reports.

Index