

Extraction des arguments d'événements à partir de peu d'exemples par méta-apprentissage

Aboubacar Tuo Romaric Besançon Olivier Ferret Julien Tourille

Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

{aboubacar.tuo, romaric.besancon, olivier.ferret, julien.tourille}@cea.fr

RÉSUMÉ

Les méthodes d'apprentissage avec peu d'exemples pour l'extraction d'événements sont développées pour réduire le coût d'annotation des données. Cependant, la plupart des études sur cette tâche se concentrent uniquement sur la détection des déclencheurs d'événements et aucune étude n'a été proposée sur l'extraction d'arguments dans un contexte de méta-apprentissage. Dans cet article, nous étudions l'extraction d'arguments d'événements avec peu d'exemples en exploitant des réseaux prototypiques et en considérant la tâche comme un problème de classification de relations. De plus, nous proposons d'améliorer les représentations des relations en injectant des connaissances syntaxiques dans le modèle par le biais de réseaux de convolution sur les graphes. Nos évaluations montrent que cette approche obtient de bonnes performances sur ACE 2005 dans plusieurs configurations avec peu d'exemples et soulignent l'importance des connaissances syntaxiques pour cette tâche.

ABSTRACT

A Meta-Learning Approach for Few-Shot Event Argument Extraction.

Few-shot learning techniques for Event Extraction are developed to alleviate the cost of data annotation. However, most studies on few-shot event extraction only focus on event trigger detection and no study has been proposed on argument extraction in a meta-learning context. In this paper, we investigate few-shot event argument extraction using prototypical networks, casting the task as a relation classification problem. Furthermore, we propose to enhance the relation embeddings by injecting syntactic knowledge into the model using graph convolutional networks. Our experimental results show that our proposed approach achieves strong performance on ACE 2005 in several few-shot configurations and highlight the importance of syntactic knowledge for this task.

MOTS-CLÉS : Extraction d'information, extraction d'événements, extraction d'arguments, apprentissage à partir de peu d'exemples.

KEYWORDS: Information extraction, event extraction, argument extraction, few-shot learning.

1 Introduction

L'extraction d'événements vise à identifier et extraire automatiquement des informations sur des événements à partir de textes non structurés, en se focalisant plus spécifiquement, pour chaque événement, sur son déclencheur (le mot ou la phrase correspondant à la mention de l'événement) et ses arguments (les entités qui jouent un rôle dans l'événement). Par exemple, dans la phrase « *Seven U.S. soldiers were killed when their vehicle hit an explosive device in Baghdad* », un événement de type Décès (*Life.Die* selon la nomenclature de référence de ACE 2005) est évoqué, associé au déclencheur

killed et aux arguments *Seven U.S. soldiers*, *explosive device* et *Baghdad*, qui correspondent aux rôles respectifs de victime, d'instrument et de lieu dans la structure de l'événement. Les systèmes d'extraction d'événements reposent classiquement sur des approches supervisées, qui nécessitent une grande quantité de données annotées pour chaque type d'événement considéré. Cette annotation étant coûteuse, elle ne peut pas être effectuée pour toutes les applications, en particulier celles pour lesquelles de nouveaux types d'événements peuvent apparaître avec seulement quelques exemples. En conséquence, un intérêt croissant s'est développé pour relever le défi de l'extraction d'événements à partir de peu d'exemples.

La plupart des études dans ce domaine se concentrent uniquement sur la détection d'événements, qui consiste à extraire et classer les déclencheurs d'événements. Plusieurs de ces travaux reposent sur l'utilisation du méta-apprentissage et des réseaux prototypiques (Cong *et al.*, 2021; Tuo *et al.*, 2022, 2023). Cependant, très peu d'études abordent l'extraction des arguments d'événements en utilisant ces méthodes par méta-apprentissage. La plupart des méthodes existantes pour l'extraction des arguments d'événements dans des scénarios à faibles ressources ne relèvent pas en effet de l'apprentissage par transfert, mais cherchent plutôt à limiter la dégradation de leurs performances en présence d'une quantité limitée de données annotées. Ces études exploitent un large ensemble de méthodes allant du question-réponse (Du & Cardie, 2020; Zhou *et al.*, 2021) à l'implication textuelle (Sainz *et al.*, 2022) en passant par les méthodes génératives (Chen *et al.*, 2020; Hsu *et al.*, 2022; Dai *et al.*, 2022; Ma *et al.*, 2022). Des approches sans données annotées (*zero-shot*) ont également été proposées, soit en s'appuyant sur des ressources externes (Huang *et al.*, 2018; Zhang *et al.*, 2021), soit en utilisant des techniques de génération avec des modèles de langue pré-entraînés (Lin *et al.*, 2023).

Dans cette étude, nous proposons d'aborder l'extraction des arguments d'événements à partir de peu d'exemples à l'aide de réseaux prototypiques. Ce choix est principalement motivé par l'efficacité démontrée de ces méthodes dans la tâche de détection d'événements et dans plusieurs autres travaux en extraction d'information (Han *et al.*, 2018; Gao *et al.*, 2019; Fritzler *et al.*, 2018; Lai *et al.*, 2021). Notre objectif est d'évaluer les capacités de ces approches prototypiques dans le cadre de l'extraction des arguments et de proposer un cadre d'évaluation adapté à cette tâche dans un contexte de méta-apprentissage. Par ailleurs, nous proposons et évaluons deux approches pour l'injection d'informations syntaxiques dans la représentation des arguments d'événement.

2 Approche

2.1 Formulation du problème

Nous abordons la tâche d'extraction des arguments d'événements comme une tâche de classification de relations entre un déclencheur événementiel et les entités de la phrase abritant ce déclencheur¹. Il s'agit plus précisément, pour chacune de ces entités, d'une classification multiclasse, chaque classe correspondant à l'un des rôles possibles pour le type d'événement associé au déclencheur considéré. S'y ajoute une classe dite *NULLE* pour signifier l'absence de rôle de l'entité candidate par rapport à l'événement.

De façon comparable à la détection d'événements, cette tâche peut être traitée selon la formulation épisodique *N*-ways, *k*-shots (Vinyals *et al.*, 2016) dans un contexte de méta-apprentissage, avec

1. Nous nous restreignons dans cette étude à la détection d'arguments d'événements au sein d'une même phrase.

cependant une légère variation. Cette formulation distingue, comme un apprentissage supervisé standard, des ensembles d'apprentissage, de validation et de test, mais à la différence du cas classique, les classes sont différentes pour ces trois ensembles. L'entraînement du modèle cible se fait en échantillonnant un grand nombre de fois un sous-ensemble de N classes au sein de l'ensemble d'apprentissage, avec k exemples par classe. Cet échantillonnage permet de créer autant de versions restreintes d'un ensemble d'apprentissage, appelé *support set*, et de test, appelé *query set*. Chaque itération dans ce cadre est appelée *épisode* et les poids du modèle sont mis à jour après chaque épisode. Cette façon de faire permet ainsi d'entraîner le modèle cible à un apprentissage par transfert, d'où le terme de méta-apprentissage.

Dans le cas de l'extraction d'arguments d'événements, bien que la classification concerne effectivement les arguments des événements, nous considérons les nouvelles classes au niveau des types d'événements. Cette approche correspond au scénario dans lequel les types d'événements de l'ensemble de test n'ont pas été rencontrés lors de l'entraînement, ce qui est davantage en phase avec les applications du monde réel où de nouveaux événements peuvent apparaître plutôt que de nouveaux rôles d'argument pour des événements existants. Par conséquent, notre formulation de l'approche N -ways, k -shots induit un N variable représentant le nombre d'arguments pour un type d'événement donné, chacune ayant k instances dans le support set. Par ailleurs, même si l'on peut considérer en toute généralité qu'un rôle n'est pas indépendant du type d'événement auquel il se rattache, des types d'événements différents peuvent contenir les mêmes rôles d'argument qui, au-delà de la similarité de nom, partagent de fait certaines similarités plus sémantiques, sans forcément être véritablement identiques². Par conséquent, certains rôles de l'ensemble de test peuvent déjà avoir été vus lors de l'entraînement du modèle, liés à un autre type d'événement.

Dans ce contexte, chaque épisode se présente comme une tâche de classification, notée $\mathcal{T} = \{\mathcal{S}, \mathcal{Q}\}$ et comportant un support set, $\mathcal{S}$, et un query set, $\mathcal{Q}$. Le support set $\mathcal{S}$ comprend un type d'événement avec ses N classes d'arguments, chacune étant représentée par k instances annotées, tandis que le query set contient une phrase mentionnant au moins un événement du même type. Dans le paradigme des approches prototypiques, la classification se fait par un encodage des exemples du support set, la construction de prototypes en combinant ces exemples (le plus souvent par une moyenne) et l'étape de classification elle-même se fait par la sélection du prototype le plus proche pour chaque exemple du query set. La figure 1 donne une vue d'ensemble de ce dispositif.

2.2 Encodage d'une instance

Pour un type d'événement e donné, une instance est définie par (x_i, y_i) avec $x_i = (s_i^e, tr_i^e, a_i)$, où s_i^e est la phrase mentionnant l'événement, tr_i^e le déclencheur, a_i le candidat argument, et y_i le rôle appartenant à $\mathcal{A}^e = \mathcal{A}_+^e \cup \{NULLE\}$, $\mathcal{A}_+^e$ étant l'ensemble des arguments du type d'événement e et $NULLE$ indiquant que l'entité n'a aucun rôle dans l'événement. La même phrase peut donc appartenir à autant d'exemples qu'elle contient de mentions d'entités.

Chaque instance est traitée par un encodeur pour produire une représentation vectorielle $h_i = \mathcal{E}(x_i)$ pour chaque paire déclencheur-entité dans le contexte d'un événement donné. En pratique, cette représentation est obtenue en concaténant les représentations du déclencheur et de l'entité résultant de leur encodage contextuel par un modèle de langue de type BERT (Devlin *et al.*, 2019). La

2. Par exemple, dans le domaine judiciaire, des événements différents comme une arrestation, une accusation ou une condamnation peuvent tous avoir un argument associé *Crime*, correspondant au crime pour lequel on est arrêté, accusé ou condamné.

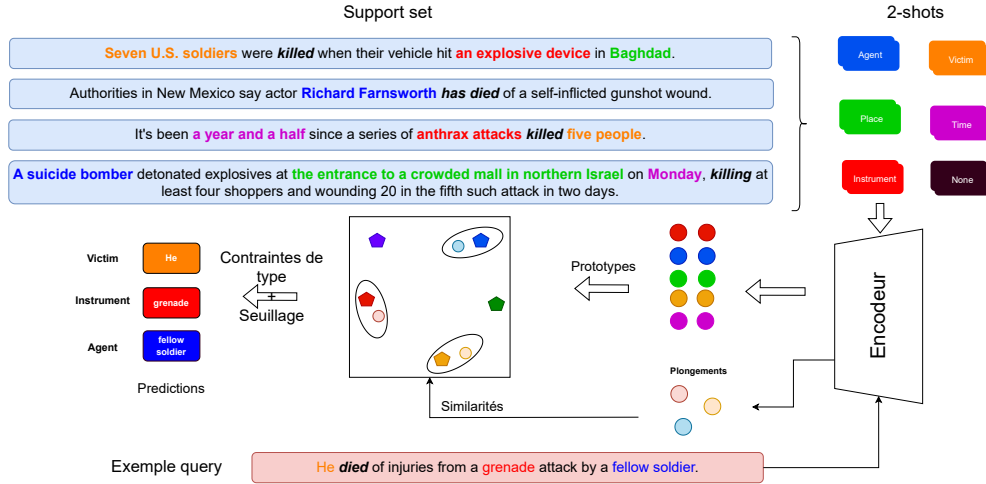


FIGURE 1 – Vue d’ensemble de notre modèle. Les déclencheurs sont en *gras italique* et chaque couleur correspond à une classe d’arguments.

représentation d’un empan de plusieurs sous-mots (*tokens*) est obtenue en prenant la moyenne des représentations de ces sous-mots. Cette représentation constitue l’entrée du classifieur, de type réseau prototypique, permettant de décider du type de relation existant entre le déclencheur et l’entité candidate (cf. section 2.4).

2.3 Intégration des informations syntaxiques

La représentation des exemples revêt une importance toute particulière dans le cas des réseaux prototypiques, car elle détermine très directement la forme des prototypes représentant chacune des classes et servant de base à la classification. Dans le cas de l’extraction d’arguments, nous proposons ainsi d’exploiter les relations syntaxiques entre le déclencheur et les entités pour aider à distinguer les entités lorsque leurs représentations (issues du modèle de langue contextuel) ne sont pas suffisamment discriminantes. Plus précisément, nous avons observé que certaines entités peuvent être confondues au sein d’un même événement, notamment lorsque leurs rôles se ressemblent ou sont symétriques. Par exemple, dans un contexte d’attaque, l’entité représentant l’agent attaquant peut parfois être confondue avec celle représentant la cible de l’attaque et inversement. De même, dans un contexte de transport, les entités représentant l’origine et la destination peuvent être sujettes à confusion. Pour lever ces ambiguïtés, nous proposons d’exploiter des informations syntaxiques additionnelles en nous appuyant sur le constat que les rôles des entités au sein d’un événement sont souvent étroitement liés à leurs rôles syntaxiques dans les phrases qui les décrivent. Dans l’exemple illustré de la figure 2, le déclencheur « fired » peut ainsi être lié à son argument « police officer » du fait de la présence d’une relation syntaxique de sujet (en voix passive) alors que ces deux éléments sont distants dans la phrase.

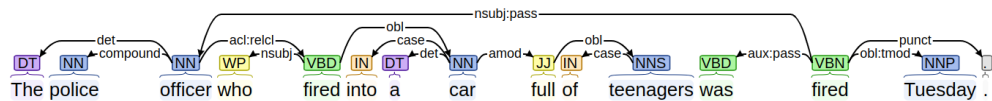


FIGURE 2 – Analyse en dépendances syntaxiques et étiquettes morphosyntaxiques pour la phrase : *The police officer who fired into a car full of teenagers was fired Tuesday.*

Nous utilisons l’encodeur BERT comme point de référence pour évaluer les avantages de l’injection d’informations syntaxiques. Sur cette base, nous proposons ensuite deux façons de prendre en compte les informations syntaxiques pour enrichir les représentations des paires déclencheur/entité.

BERT++ intègre de manière statique les informations syntaxiques en associant un vecteur à chaque type de relation syntaxique, ces vecteurs étant ajustés pendant l’apprentissage du modèle. Plus précisément, cette approche consiste à concaténer les plongements contextuels du déclencheur et de l’entité avec les représentations vectorielles des étiquettes morphosyntaxiques de l’entité et du chemin de dépendances syntaxiques entre le déclencheur et l’entité. La longueur des chemins de dépendances syntaxiques étant variable, nous avons appliqué une agrégation *max-pooling* sur l’ensemble des étiquettes de dépendance syntaxique afin d’obtenir les plongements de ces chemins.

BERT-GCN combine les informations syntaxiques avec les plongements fournis par BERT en utilisant un réseau de convolution sur les graphes (GCN) (Kipf & Welling, 2017). Cette approche est particulièrement intéressante dans notre cas, car elle permet de différencier les relations syntaxiques, contribuant ainsi à réduire les confusions entre des arguments similaires. Cependant, cette modélisation entraîne une augmentation significative du nombre de paramètres à apprendre, correspondant aux paramètres du GCN.

Afin de permettre à l’encodeur BERT de capturer des informations sur l’emplacement du déclencheur et de l’entité cible au sein d’une phrase, nous marquons le début et la fin de ces éléments avec des tokens spéciaux, comme suggéré dans d’autres travaux de la littérature (Zhang *et al.*, 2019; Han *et al.*, 2018; Baldini Soares *et al.*, 2019). La phrase est ensuite fournie à l’encodeur BERT pour en obtenir une représentation de chaque token. Ces représentations sont ensuite utilisées pour initialiser les nœuds de l’arbre de dépendances syntaxiques qui sert d’entrée au réseau de convolution. Nous introduisons un nouveau type de relation syntaxique appelé *subword* correspondant au fait qu’un mot puisse être découpé en plusieurs tokens par le tokeniseur de BERT. Après apprentissage du réseau de convolution sur les graphes, la représentation de chaque empan est obtenue en prenant la moyenne des représentations des nœuds appartenant à cet empan.

2.4 Classification des instances

Notre modèle reposant sur les réseaux prototypiques, la classification des instances est réalisée en fonction de leur similarité avec les représentations des prototypes de chaque classe. À cette fin, nous avons mené des expériences avec deux approches : les réseaux prototypiques standard (*Prototypical Networks*, **Proto**) et leur version contrastive (*Contrastive Prototypical Networks*, **C-Proto**), proposée par Tuo *et al.* (2023).

La différence entre ces deux modèles réside dans le fait que **C-Proto** adopte un apprentissage contrastif et ne construit pas de prototype pour la classe *NULLE*, qui regroupe les entités n’ayant pas de rôle dans les événements. Dans ce cas, les exemples de cette classe sont filtrés à l’aide d’un seuil de similarité. Alors que le modèle **Proto** s’appuie sur une fonction de coût de type entropie croisée, couplée à une fonction softmax, le modèle **C-Proto** est entraîné avec une fonction de coût de charnière (*hinge loss*). Cette dernière est plus précisément la somme de deux termes : l’un ($\mathcal{L}_+$) concerne uniquement les arguments et l’autre ($\mathcal{L}_-$), les entités non-arguments (cf. équations 1 et 2). Nous notons $\mathcal{Q}^+$ le sous-ensemble des arguments du query set d’un épisode et $\mathcal{Q}^-$, le sous-ensemble du query set contenant des entités non-arguments).

$$\mathcal{L}_+(\mathcal{S}, \mathcal{Q}) = \sum_{(x_i, y_i) \in \mathcal{Q}^+} \sum_{j \neq y_i} \max_j(0, \mathcal{M}_0 - s(h_i, c^j) + s(h_i, c^{y_i})) \quad (1)$$

$$\mathcal{L}_-(\mathcal{S}, \mathcal{Q}) = \max_{(x_i, y_i) \in \mathcal{Q}^+} (0, \max_{x_j \in \mathcal{Q}^-} (s(h_j, c^{y_i}) - \mathcal{M}_1)) \quad (2)$$

où $s(., .)$ est une fonction de similarité, c^i le prototype de la classe i , h_i la représentation dense d'une entité, $\mathcal{M}_0$ et $\mathcal{M}_1$ des hyperparamètres permettant de contrôler la marge entre les arguments et les entités non-arguments. Une telle marge laisse ensuite la possibilité de fixer un seuil de similarité permettant de séparer les arguments des entités non-arguments.

Contrairement à [Tuo et al. \(2023\)](#), qui ont recours à une fonction de répartition pour estimer ce seuil, nous calculons ici le seuil en utilisant la valeur de similarité trouvée pour l'exemple le plus proche dans le support set. En effet, dans le cas de l'extraction des arguments, certaines phrases peuvent en effet ne contenir qu'une seule entité candidate ou seulement des entités correspondant à des non-arguments³. Par conséquent, l'utilisation de la fonction de répartition dans le cas d'une seule entité ou d'entités n'ayant pas de rôle ne permettrait pas de fixer un seuil raisonnable. Il faudrait, dans ce cas, que le seuil soit supérieur aux similarités pour toutes les entités.

De plus, à l'instar de travaux antérieurs sur l'extraction d'événements ([Sainz et al., 2022](#); [Lin et al., 2023](#)), nous utilisons la connaissance préalable des types des entités pour contraindre les prédictions des rôles d'arguments et améliorer ainsi la précision des prédictions.

En pratique, lors de l'évaluation du modèle, nous commençons par vérifier la compatibilité entre le rôle prédit et le type de l'entité candidate. Si la contrainte n'est pas satisfaite, nous considérons la prédiction comme incorrecte et prenons la classe du prototype suivant le plus proche jusqu'à ce que le rôle prédit corresponde au type d'entité ou à la classe *NULLE*, qui n'est soumise à aucune contrainte.

3 Expériences

3.1 Paramètres expérimentaux

Nous avons mené nos expériences sur l'ensemble de données ACE-2005 ([Walker et al., 2006](#)), avec la partition fournie par [Lai et al. \(2020\)](#). Cette partition garantit qu'il n'y a pas de chevauchement entre les types d'événements dans les ensembles d'entraînement et d'évaluation, simulant ainsi un scénario réaliste avec une faible disponibilité des données. Nous donnons plus de détails sur ce jeu de données à l'Annexe A.

Nous utilisons un encodeur **BERT** pour fournir les représentations des mots à partir des phrases en entrée. De plus, pour l'encodeur **BERT++**, nous utilisons des vecteurs entraînaables de taille 256 pour encoder les dépendances syntaxiques et les étiquettes morphosyntaxiques, obtenues grâce à l'analyseur spaCy. Pour **BERT-GCN**, le nombre de couches de convolution du GCN a été fixé à 2, car c'est ce qui donne le meilleur résultat empirique sur l'ensemble de développement.

3. Dans ces cas, les arguments de la mention d'événement considérée se trouvent souvent dans d'autres phrases. Il faudrait une extraction au niveau du document pour les identifier.

3.2 Résultats

Nous reportons nos principaux résultats dans le tableau 1. Les entités considérées pour l’extraction des arguments sont les entités annotées dans le jeu de données.

Encodeur	Modèle	5-shots			10-shots		
		P	R	F1	P	R	F1
BERT	Proto	63,1 ± 0,9	56,4 ± 1,0	59,6 ± 0,5	66,4 ± 0,5	61,6 ± 0,7	63,9 ± 0,3
	C-Proto	62,7 ± 0,9	57,0 ± 1,2	60,0 ± 1,0	<u>67,1 ± 0,8</u>	<u>63,8 ± 0,9</u>	<u>65,5 ± 0,8</u>
BERT++	Proto	64,9 ± 1,1	58,6 ± 1,2	61,6 ± 0,8	66,8 ± 1,5	63,8 ± 1,1	65,2 ± 0,6
	C-Proto	65,8 ± 0,5	<u>58,8 ± 1,8</u>	<u>62,1 ± 1,0</u>	66,8 ± 1,7	66,5* ± 1,7	66,7* ± 1,0
BERT-GCN	Proto	69,0 ± 2,1	56,6 ± 4,0	62,2 ± 2,2	71,2* ± 0,7	60,0 ± 1,5	65,0 ± 0,9
	C-Proto	<u>68,5 ± 1,1</u>	59,2* ± 1,7	63,5* ± 1,2	69,2 ± 0,5	61,4 ± 0,8	65,1 ± 0,5

TABLE 1 – Résultats de l’extraction des arguments d’événements : Précision (P), Rappel (R) et F1-mesure (F1). Nos meilleurs scores sont en **gras**, les deuxièmes meilleurs scores sont soulignés. * indique que le meilleur score est statistiquement significatif par rapport au deuxième.

Ces résultats nous permettent de tirer deux conclusions principales. D’une part, quel que soit l’encodeur considéré, nous pouvons observer que la version contrastive **C-Proto** affiche des performances légèrement supérieures à celles du réseau prototypique standard. Cette observation vient confirmer les constatations faites par [Tuo et al. \(2023\)](#) sur l’efficacité de cette approche, en particulier en ce qui concerne la gestion de la classe *NULLE*. Toutefois, les apports dans ce cadre sont bien moindres que ceux rapportés pour la tâche de détection d’événements. Cela peut être lié à la difficulté relative de la tâche d’extraction des arguments par rapport à l’extraction des déclencheurs. En effet, les entités occupant un même rôle n’ont pas forcément de similarités sémantiques entre elles, contrairement aux déclencheurs, qui appartiennent souvent au même champ lexical.

D’autre part, nous voyons que l’intégration des informations syntaxiques améliore les performances dans tous les cas. Cette constatation suggère que l’exploitation d’informations syntaxiques pour enrichir la relation entre les déclencheurs d’événements et leurs arguments est bien bénéfique pour la tâche d’extraction des arguments d’événements. Par ailleurs, l’intégration dynamique **BERT-GCN** semble plus efficace que l’intégration statique **BERT++** lorsque très peu de données sont disponibles (5-shots).

Nous comparons de façon plus précise nos trois encodeurs à la figure 3, où nous donnons la moyenne de la F1-mesure pour chaque rôle. Dans l’ensemble, ces résultats montrent que l’intérêt des informations syntaxiques est observé à la fois pour les rôles vus pendant l’entraînement (c’est-à-dire des rôles qui portent le même nom, mais qui correspondent à des types d’événements différents) et pour les nouveaux rôles apparaissant uniquement pendant l’évaluation. Les apports se manifestent particulièrement pour les rôles pour lesquels l’encodeur **BERT** standard montre des performances relativement faibles. En revanche, pour des rôles considérés comme « faciles », qui ne portent pas à confusion, comme *Instrument*, *Vehicule* ou *Money*, l’encodeur **BERT** standard demeure très compétitif et les informations syntaxiques ont un impact moins significatif, voire parfois négatif. En particulier, l’encodeur **BERT-GCN** semble contribuer principalement à équilibrer les performances pour les rôles qui peuvent être confondus, tels que *Origin* et *Destination* ou encore *Buyer* et *Seller*. En revanche, il nuit moins aux rôles non ambigus, tels que *Instrument* ou *Vehicule*.

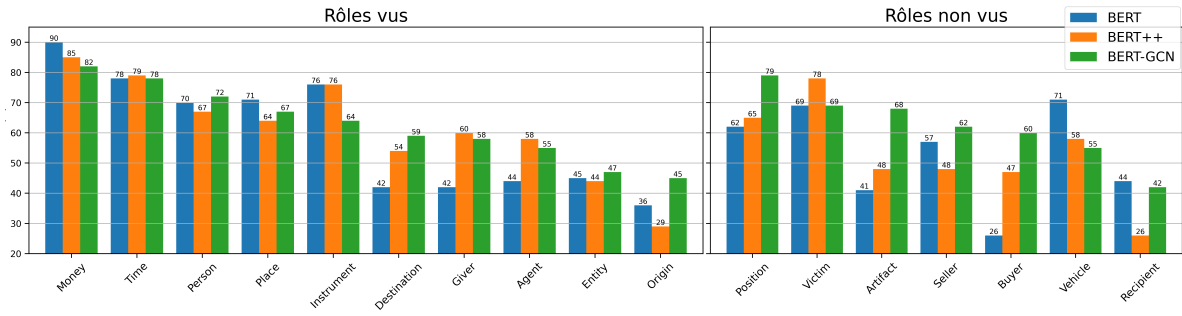


FIGURE 3 – Moyenne de la F1-mesure par rôle pour les trois encodeurs. À gauche, les rôles vus pendant l’entraînement des modèles et à droite, les rôles vus seulement pendant l’évaluation.

	P	R	F1
Modèle entier	68,5	59,2	63,5
- seuillage	47,9	59,3	52,9
- contraintes	68,2	50,9	58,3
- seuillage & contraintes	33,9	61,9	43,8

TABLE 2 – Étude d’ablation pour chaque composante du modèle **C-Proto** dans une configuration 5-shots. Précision (P), Rappel (R) et F1-mesure (F1) en moyenne sur cinq expérimentations.

La figure 4 donne pour sa part un aperçu des représentations élaborées par chaque encodeur sur l’ensemble d’évaluation. L’encodeur **BERT pré-entraîné** correspond à un modèle BERT sans aucun ajustement spécifique à la tâche d’extraction des arguments d’événements. Nous comparons les trois encodeurs présentés dans ce travail : **BERT**, **BERT++** et **BERT-GCN**.

Tout d’abord, il est évident que l’entraînement du modèle BERT améliore notablement le caractère discriminant des plongements du point de vue des classes d’arguments par rapport à un BERT non affiné. Cela met en lumière la pertinence de la formulation que nous avons adoptée pour cette tâche et l’importance de l’affinage du modèle BERT dans ce contexte.

On peut également observer qualitativement que les deux encodeurs enrichis, **BERT++** et **BERT-GCN**, semblent fournir des représentations plus discriminantes que l’encodeur BERT d’origine. Ces observations correspondent aux résultats obtenus lors de l’évaluation, ce qui renforce la pertinence de l’enrichissement des représentations par des informations syntaxiques et suggère une amélioration globale de la performance du modèle.

Étude d’ablation Pour compléter notre analyse des résultats, nous présentons dans le tableau 2 une étude d’ablation réalisée avec le modèle **C-Proto**. L’objectif est d’explorer les effets de l’utilisation du seuillage et de l’introduction des contraintes liées au type des entités et à leurs rôles. Afin d’éliminer l’utilisation du seuil, nous avons reconstruit un prototype pour la classe nulle lors de la phase d’évaluation.

Les résultats obtenus dans cette étude d’ablation mettent en évidence l’utilité de chacune de ces opérations sur les performances globales du modèle. En général, nous observons que l’utilisation du

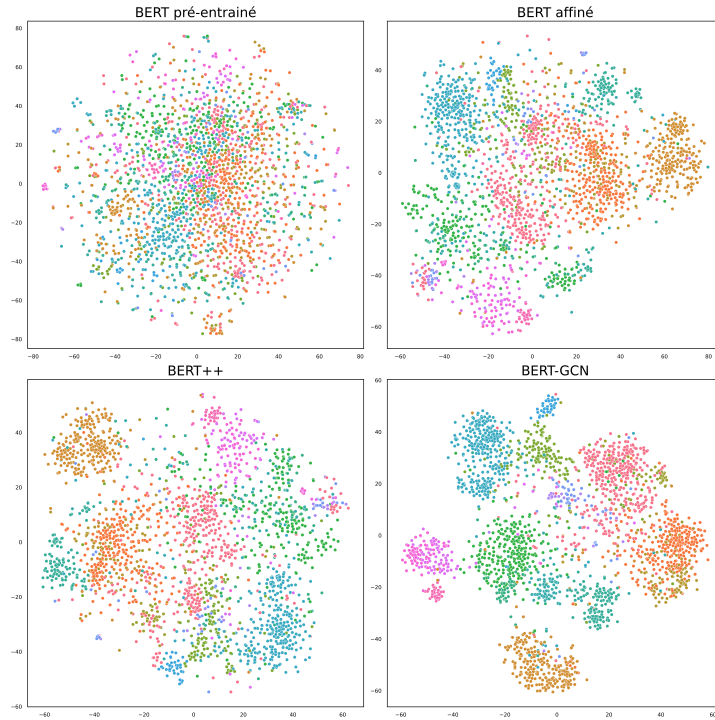


FIGURE 4 – Visualisation des représentations des arguments en utilisant la méthode t-SNE. Chaque point correspond à un argument et la couleur, à une classe de rôle.

seuillage conduit principalement à une amélioration de la précision, bien qu’elle soit accompagnée d’une légère diminution du rappel en contrepartie. Il faut noter que l’introduction du seuillage visait précisément à réduire le taux de faux positifs et, donc, à une augmentation de la précision.

Les contraintes relatives aux types des entités et leurs rôles ont aussi un impact positif sur les performances, bien que leur contribution soit plus modeste par rapport au seuillage. Ces contraintes visent principalement à réduire les confusions entre certains arguments. Comme mentionné précédemment, il arrive que des entités de même type soient parfois confondues dans leurs rôles et, bien que ce filtrage n’élimine pas entièrement cette ambiguïté, il contribue néanmoins à la résoudre partiellement.

Comparaison avec l’état de l’art Comme nous l’indiquions en introduction, la comparaison avec l’état de l’art n’est pas évidente dans la mesure où les méthodes d’évaluation pour les approches par méta-apprentissage ne sont pas directement comparables aux évaluations menées par les travaux existants en extraction d’arguments d’événements, qui reprennent le paradigme classique de l’apprentissage supervisé mais dans une configuration dégradée de faible quantité de données annotées.

Pour réaliser une forme de comparaison, nous nous sommes focalisés sur la quantité de données annotées concernant les types d’événements de l’ensemble de test du jeu de données ACE 2005. La figure 5 fait ainsi apparaître l’évolution des performances pour l’extraction d’arguments d’événements pour trois modèles de référence – **PAIE** (Ma *et al.*, 2022), **BIP** (Dai *et al.*, 2022) et **NLI** (Sainz *et al.*, 2022) – en fonction du pourcentage des données d’entraînement. Au regard de ces courbes, nous avons fait figurer le niveau de performance de notre configuration 5-shots (c’est-à-dire 5 exemples par rôle), qui correspond en quantité à environ 3 % des données d’évaluation. Il faut toutefois noter que l’entraînement de notre modèle se fait sur 18 types d’événements et tous leurs exemples, les 3 %

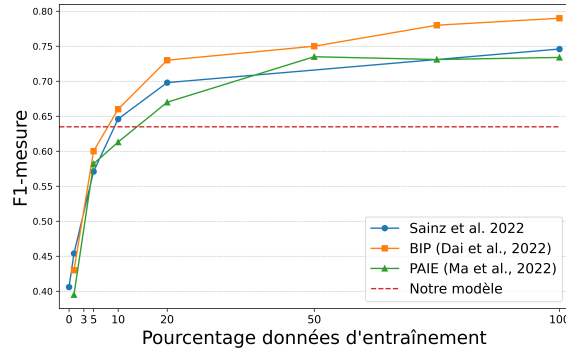


FIGURE 5 – Comparaison avec des modèles de l’état de l’art avec peu de données annotées.

de données ne concernant que les types non vus pendant l’entraînement. Néanmoins, la figure 5 fait apparaître que jusqu’à 5 % des données d’entraînement, le modèle que nous proposons obtient des performances supérieures aux modèles de référence considérés. Nous pouvons donc en conclure que notre approche par méta-apprentissage est particulièrement adaptée à un régime de très faible quantité de données annotées pour les types d’événements cibles.

4 Conclusion et perspectives

Nous avons exploré l’extraction des arguments d’événements dans un scénario de faible disponibilité de données pour définir de nouvelles classes en mettant l’accent sur l’utilisation d’informations syntaxiques pour lever l’ambiguïté concernant certains rôles événementiels. Nous avons repensé ce problème en le formulant comme une tâche de classification de relations entre le déclencheur et les arguments et en le traitant dans un cadre de méta-apprentissage à l’aide de réseaux prototypiques. Dans ce contexte, nous avons adapté plus précisément le cadre de classification N -ways, k -shots pour répondre aux besoins spécifiques de l’extraction d’arguments d’événements. Les évaluations menées ont montré l’intérêt de la prise en compte de ces informations syntaxiques, en particulier pour les rôles les plus susceptibles de se confondre.

Le travail mené fait l’hypothèse de la connaissance a priori des déclencheurs événementiels et des entités candidates pour les rôles et présuppose donc une architecture de type pipeline, qui présente l’avantage d’une certaine modularité mais souffre de problèmes connus de propagation d’erreurs entre modules. À l’instar des travaux de [Nguyen et al. \(2021\)](#), dans un contexte de quantités importantes de données annotées, un des prolongements naturels de notre travail sera d’examiner comment des approches jointes impliquant l’extraction d’entités nommées, de déclencheurs événementiels et d’arguments d’événements peuvent être mises en œuvre dans un contexte de faible quantité de données annotées disponibles par le biais de méthodes de méta-apprentissage.

Remerciements Ces travaux ont été réalisés grâce au supercalculateur Factory-IA financé par le Conseil Régional d’Île-de-France.

Références

- BALDINI SOARES L., FITZGERALD N., LING J. & KWIATKOWSKI T. (2019). Matching the blanks : Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 2895–2905, Florence, Italy : Association for Computational Linguistics. DOI : [10.18653/v1/P19-1279](https://doi.org/10.18653/v1/P19-1279).
- CHEN Y., CHEN T., EBNER S., WHITE A. S. & VAN DURME B. (2020). Reading the manual : Event extraction as definition comprehension. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, p. 74–83, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2020.spnlp-1.9](https://doi.org/10.18653/v1/2020.spnlp-1.9).
- CONG X., CUI S., YU B., LIU T., YUBIN W. & WANG B. (2021). Few-Shot Event Detection with Prototypical Amortized Conditional Random Field. In *Findings of the Association for Computational Linguistics : ACL-IJCNLP 2021*, p. 28–40, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2021.findings-acl.3](https://doi.org/10.18653/v1/2021.findings-acl.3).
- DAI L., WANG B., XIANG W. & MO Y. (2022). Bi-directional iterative prompt-tuning for event argument extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, p. 6251–6263, Abu Dhabi, United Arab Emirates : Association for Computational Linguistics.
- DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423).
- DU X. & CARDIE C. (2020). Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 671–683, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2020.emnlp-main.49](https://doi.org/10.18653/v1/2020.emnlp-main.49).
- FRITZLER A., LOGACHEVA V. & KRETOV M. (2018). Few-shot classification in Named Entity Recognition Task. *arXiv :1812.06158 [cs, stat]*. arXiv : 1812.06158, DOI : [10.1145/3297280.3297378](https://doi.org/10.1145/3297280.3297378).
- GAO T., HAN X., LIU Z. & SUN M. (2019). Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, **33**, 6407–6414. DOI : [10.1609/aaai.v33i01.33016407](https://doi.org/10.1609/aaai.v33i01.33016407).
- HAN X., ZHU H., YU P., WANG Z., YAO Y., LIU Z. & SUN M. (2018). FewRel : A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 4803–4809, Brussels, Belgium : Association for Computational Linguistics. DOI : [10.18653/v1/D18-1514](https://doi.org/10.18653/v1/D18-1514).
- HSU I.-H., HUANG K.-H., BOSCHÉ E., MILLER S., NATARAJAN P., CHANG K.-W. & PENG N. (2022). DEGREE : A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1890–1908, Seattle, United States : Association for Computational Linguistics. DOI : [10.18653/v1/2022.naacl-main.138](https://doi.org/10.18653/v1/2022.naacl-main.138).
- HUANG L., JI H., CHO K., DAGAN I., RIEDEL S. & VOSS C. (2018). Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2160–2170, Melbourne, Australia : Association for Computational Linguistics. DOI : [10.18653/v1/P18-1201](https://doi.org/10.18653/v1/P18-1201).

- KIPF T. N. & WELLING M. (2017). Semi-supervised classification with graph convolutional networks.
- LAI V., DERNONCOURT F. & NGUYEN T. H. (2021). Learning Prototype Representations Across Few-Shot Tasks for Event Detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 5270–5277, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics.
- LAI V. D., NGUYEN T. H. & DERNONCOURT F. (2020). Extensively matching for few-shot learning event detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, p. 38–45, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2020.nuse-1.5](https://doi.org/10.18653/v1/2020.nuse-1.5).
- LIN Z., ZHANG H. & SONG Y. (2023). Global constraints with prompting for zero-shot event argument classification. In *Findings of the Association for Computational Linguistics : EACL 2023*, p. 2527–2538, Dubrovnik, Croatia : Association for Computational Linguistics.
- MA Y., WANG Z., CAO Y., LI M., CHEN M., WANG K. & SHAO J. (2022). Prompt for extraction ? PAIE : Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 6759–6774, Dublin, Ireland : Association for Computational Linguistics. DOI : [10.18653/v1/2022.acl-long.466](https://doi.org/10.18653/v1/2022.acl-long.466).
- NGUYEN M. V., LAI V. D. & NGUYEN T. H. (2021). Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 27–38, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2021.naacl-main.3](https://doi.org/10.18653/v1/2021.naacl-main.3).
- SAINZ O., GONZALEZ-DIOS I., LOPEZ DE LACALLE O., MIN B. & AGIRRE E. (2022). Textual entailment for event argument extraction : Zero- and few-shot with multi-source learning. In *Findings of the Association for Computational Linguistics : NAACL 2022*, p. 2439–2455, Seattle, United States : Association for Computational Linguistics. DOI : [10.18653/v1/2022.findings-naacl.187](https://doi.org/10.18653/v1/2022.findings-naacl.187).
- TUO A., BESANÇON R., FERRET O. & TOURILLE J. (2022). Better exploiting bert for few-shot event detection. In P. ROSSO, V. BASILE, R. MARTÍNEZ, E. MÉTAIS & F. MEZIANE, Édts., *Natural Language Processing and Information Systems*, p. 291–298, Cham : Springer International Publishing.
- TUO A., BESANÇON R., FERRET O. & TOURILLE J. (2023). Trigger or not trigger : Dynamic thresholding for few shot event detection. In J. KAMPS, L. GOEURLOT, F. CRESTANI, M. MAISTRO, H. JOHO, B. DAVIS, C. GURRIN, U. KRUSCHWITZ & A. CAPUTO, Édts., *45th European Conference on Information Retrieval (ECIR 2023) : Advances in Information Retrieval, short article session*, volume 13981 de *Lecture Notes in Computer Science*, p. 637–645, Dublin, Ireland : Springer Nature Switzerland.
- VINYALS O., BLUNDELL C., LILLICRAP T., KAVUKCUOGLU KORAY K. & WIERSTRA D. (2016). Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29.
- WALKER C., STRASSEL S. & JULIE MEDERO K. M. (2006). *ACE 2005 Multilingual Training Corpus*. LDC corpora. Linguistic Data Consortium. DOI : [10.35111/mwxc-vh88](https://doi.org/10.35111/mwxc-vh88).
- ZHANG H., WANG H. & ROTH D. (2021). Zero-shot Label-aware Event Trigger and Argument Classification. In *Findings of the Association for Computational Linguistics : ACL-IJCNLP 2021*, p. 1331–1340, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2021.findings-acl.114](https://doi.org/10.18653/v1/2021.findings-acl.114).

ZHANG Z., HAN X., LIU Z., JIANG X., SUN M. & LIU Q. (2019). ERNIE : Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 1441–1451, Florence, Italy : Association for Computational Linguistics. DOI : [10.18653/v1/P19-1139](https://doi.org/10.18653/v1/P19-1139).

ZHOU Y., CHEN Y., ZHAO J., WU Y., XU J. & LI J. (2021). What the role is vs. what plays the role : Semi-supervised event argument extraction via dual question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**(16), 14638–14646. DOI : [10.1609/aaai.v35i16.17720](https://doi.org/10.1609/aaai.v35i16.17720).

A Jeu de données ACE-2005

ACE-2005 est un jeu de données de référence pour l'extraction d'information, composé de 599 documents⁴ de diverses sources, en anglais, mandarin et arabe. Dans sa version anglaise, il propose huit types d'événements, subdivisés en 33 sous-types et 34 rôles pour les arguments. Sa richesse et sa diversité en font une ressource clé pour le développement de modèles d'extraction d'information multilingues et polyvalents. Nous listons l'ensemble des types et sous-types dans le tableau 3.

Pour nos expérimentations, nous adoptons le découpage de [Lai et al. \(2020\)](#), qui considère quatre types (*Conflict*, *Business*, *Contact* et *Justice*) dans l'ensemble d'entraînement et les autres types dans l'ensemble d'évaluation. Dans ce découpage, certains rôles n'apparaissent que dans l'ensemble d'évaluation, tandis que d'autres sont exclusivement présents dans l'ensemble d'entraînement ; mais il existe également des rôles communs aux deux ensembles (voir tableau 4). Nous avons supprimé les rôles d'argument comportant moins de 10 exemples afin de disposer de suffisamment de données pour constituer des épisodes avec au moins 10 exemples dans les ensembles de support. Nous avons également retiré les types pour lesquels il ne restait aucun argument (c'est-à-dire *Business:End-Org*, *Justice:Pardon*, *Justice:Extradite* et *Justice:Acquit* dans l'ensemble d'entraînement, ainsi que *Personnel:Nominate* dans l'ensemble d'évaluation).

4. Dans la version anglaise.

Types	Sous-types	Arguments
Life	Be-born	Person, Place, Time
	Marry	Person, Place, Time
	Divorce	Person, Place, Time
	Injure	Agent, Victim, Instrument, Place, Time
	Die	Agent, Victim, Instrument, Place, Time
Movement	Transport	Agent, Artifact, Vehicle, Price, Origin, Destination, Time
Transaction	Transfer-Ownership	Buyer, Seller, Beneficiary, Artifact, Price, Place, Time
	Transfer-Money	Giver, Recipient, Beneficiary, Money, Place, Time
Business	Start-Org	Agent, Org, Place, Time
	Merge-Org	Org, Place, Time
	Declare-Bankruptcy	Org, Place, Time
	End-Org	Org, Place, Time
Conflict	Attack	Attacker, Target, Instrument, Place, Time
	Demonstrate	Entity, Place, Time
Contact	Meet	Entity, Place, Time
	Phone-Write	Entity, Time
Personnel	Start-Position	Person, Entity, Position, Place, Time
	End-Position	Person, Entity, Position, Place, Time
	Nominate	Person, Agent, Position, Place, Time
	Elect	Person, Entity, Position, Place, Time
Justice	Arrest-Jail	Person, Agent, Crime, Place, Time
	Release-Parole	Person, Entity, Crime, Place, Time
	Trial-Hearing	Defendant, Prosecutor, Adjudicator, Crime, Place, Time
	Charge-Indict	Defendant, Prosecutor, Adjudicator, Crime, Place, Time, Sentence
	Sue	Plaintiff, Defendant, Adjudicator, Crime, Place, Time
	Convict	Defendant, Adjudicator, Crime, Place, Time
	Sentence	Defendant, Adjudicator, Crime, Sentence, Place, Time
	Fine	Entity, Adjudicator, Money, Crime, Place, Time
	Execute	Person, Agent, Crime, Place, Time
	Extradite	Agent, Person, Destination, Origin, Crime, Time
	Acquit	Defendant, Adjudicator, Crime, Place, Time
	Appeal	Defendant, Prosecutor, Adjudicator, Crime, Place, Time
	Pardon	Defendant, Adjudicator, Crime, Place, Time

TABLE 3 – Liste des types d'événements du jeu de données ACE-2005.

Ensemble	Arguments
Entraînement	Org, Adjudicator, Prosecutor, Defendant, Sentence, Plaintiff, Attacker, Target, Crime
Évaluation	Buyer, Seller, Recipient, Artifact, Vehicle, Position, Victim
Partagés	Agent, Person, Time, Place, Giver, Money, Entity, Instrument, Destination, Origin

TABLE 4 – Répartition des rôles d’arguments entre l’ensemble d’entraînement et l’ensemble d’évaluation du jeu de données ACE-2005.

B Synthèse des hyperparamètres

Nous donnons la liste des hyperparamètres dans le Tableau 5 ci-dessous.

Paramètre	Valeur
encodeur	BERT-large-uncased
taille des séquences	128
nombre d’itérations d’entraînement	5 000
optimiseur	AdamW
taux d’apprentissage	
BERT	$1e - 5$
autres	$1e - 4$
weight decay	$1e - 2$
dropout	0, 1
warmup ratio	0, 1
scheduler	stepLR
β_1 β_2	0,9 0,999
nombre de couches de convolution	2

TABLE 5 – Liste et valeurs des hyperparamètres.