

L'importance des entités pour la tâche de détection d'événements en tant que système de question-réponse

Emanuela Boros¹ Jose G. Moreno^{1,2} Antoine Doucet¹

(1) La Rochelle Université, L3i, 17000 La Rochelle, France

(2) Université de Toulouse, IRIT UMR 5505 CNRS, 31000 Toulouse, France

{emanuela.boros, antoine.doucet}@univ-lr.fr, jose.moreno@irit.fr

RÉSUMÉ

Dans cet article, nous abordons un paradigme récent et peu étudié pour la tâche de détection d'événements en la présentant comme un problème de question-réponse avec possibilité de réponses multiples et le support d'entités. La tâche d'extraction des déclencheurs d'événements est ainsi transformée en une tâche d'identification des intervalles de réponse à partir d'un contexte, tout en se concentrant également sur les entités environnantes. L'architecture est basée sur un modèle de langage pré-entraîné et finement ajusté, où le contexte d'entrée est augmenté d'entités marquées à différents niveaux, de leurs positions, de leurs types et, enfin, de leurs rôles d'arguments. Nos expériences sur le corpus ACE 2005 démontrent que le modèle proposé exploite correctement les informations sur les entités dans le cadre de la détection des événements et qu'il constitue une solution viable pour cette tâche. De plus, nous démontrons que notre méthode, avec différents marqueurs d'entités, est particulièrement capable d'extraire des types d'événements non vus dans des contextes d'apprentissage en peu de coups.

ABSTRACT

Exploring Entities in Event Detection as Question Answering.

In this paper, we approach a recent and under-researched paradigm for the task of event detection by casting it as a question-answering problem with the possibility of multiple answers and the support of entities. The extraction of event triggers is, thus, transformed into the task of identifying answer spans from a context, while also focusing on the surrounding entities. The architecture is based on a pre-trained and fine-tuned language model, where the input context is augmented with entities marked at different levels, their positions, their types, and, finally, their argument roles. Experiments on the ACE 2005 corpus demonstrate that the proposed model properly leverages entity information in detecting events and that it is a viable solution for this task. Moreover, we demonstrate that our method with different entity markers is particularly able to extract unseen event types in few-shot learning settings.

MOTS-CLÉS : Détection d'événements, systèmes de question-réponse, apprentissage en peu de coups.

KEYWORDS: Event detection, question answering, few-shot learning.

1 Introduction

Dans cet article, nous abordons la tâche de détection d'événements en étudiant l'utilisation des entités dans un paradigme récent et peu étudié de la tâche de détection d'événements, qui la présente comme un problème de question-réponse. Contrairement à ces méthodes souvent basées sur l'usage de *Transformer*, où la détection d'événements et d'arguments était considérée comme une tâche de classification, un nouveau paradigme a été introduit (Du & Cardie, 2020; Liu *et al.*, 2020) formulant la tâche comme une tâche de question-réponse (QA) et de compréhension automatique de lecture (MRC), où les événements peuvent être extraits en répondant aux questions 5W1H en anglais (qui a fait quoi, quand, où, pourquoi et comment ¹).

La détection d'événements en tant que système de question-réponse Dans le travail récemment proposé par Du & Cardie (2020), les auteurs ont introduit un nouveau paradigme pour l'extraction d'événements en le formulant comme une tâche de question-réponse, qui extrait les déclencheurs et les arguments des événements de bout en bout. Pour la détection de l'événement, ils ont considéré une approche basée sur BERT qui est habituellement appliquée aux données séquentielles. La tâche de détection d'événements est basée sur la classification où les auteurs ont conçu des modèles fixes simples, tels que *qui est le déclencheur, trigger, action, verb*, sans spécifier le type d'événement. Par exemple, si le modèle *verb* est choisi, la séquence d'entrée correspondrait à : [CLS] *verb* [SEP] phrase [SEP]. Ensuite, ils utilisent un BERT séquentiel à réglage fin pour détecter les candidats au déclenchement d'événements. Parallèlement, un autre article récent (Liu *et al.*, 2020) aborde aussi la tâche d'extraction d'événements comme une tâche de réponse à des questions (QA), de manière similaire à la méthode proposée par Du & Cardie (2020). En revanche, la tâche est basée sur la classification au lieu de la méthode de QA basée sur les intervalles. Pour l'extraction de déclencheurs, il faut représenter le mot clé [EVENT] avec la phrase respective afin de calculer une représentation conjointe, comme dans l'approche proposée par Du & Cardie (2020) où le mot clé était *verb* ou *trigger*.

Détection d'événements avec informations sur les entités Dans le contexte de la détection d'événements, certains travaux ont utilisé des entités standard de différentes manières. De meilleurs résultats peuvent être obtenus avec des types d'entités standard (Nguyen & Grishman, 2015), en concaténant des embeddings initialisés de manière aléatoire pour les types d'entités. Un réseau neuronal de graphes basé sur des arbres de dépendance (Nguyen & Grishman, 2018) a également été proposé pour effectuer la détection d'événements avec une méthode de pooling qui repose sur l'agrégation des mentions d'entités. Les arguments ont fourni des indices importants pour cette tâche dans le mécanisme d'attention supervisée proposé par Liu *et al.* (2017a) pour exploiter explicitement les informations sur les arguments. D'autres méthodes qui ont tiré parti de l'information sur les arguments étaient des approches basées sur des architectures jointes (Liu *et al.*, 2018; Li *et al.*, 2019). Moreno *et al.* (2021) a également exploré l'utilisation de modèles neuronaux pré-entraînés dans le problème de la validation des relations en utilisant explicitement une représentation en triplets de phrases avec des entités marquées, prouvant que les performances d'extraction de relations pouvaient être encore améliorées en utilisant ces informations supplémentaires. En outre, Boros *et al.* (2021) a également proposé l'utilisation de modèles neuronaux pré-entraînés dans une architecture basée sur la classification BERT pour la détection d'événements.

1. 5W est défini ainsi sur <https://fr.wikipedia.org/wiki/QOQCCP> et *comment* correspond à 1H.

Bien que ces avancées récentes prétendent transformer la détection d'événements en tâche de QA (Du & Cardie, 2020; Liu *et al.*, 2020), ces modèles restent formulés comme un problème de classification séquentielle qui vise à détecter les déclencheurs d'événements de types spécifiques. Dans cet article, nous nous concentrons sur la tâche de détection d'événements comme une tâche de QA, mais avec la possibilité de réponses multiples dans le cas où plus d'un événement est présent dans le texte.

2 Modèle

Notre modèle étend le modèle pré-entraîné de BERT (Devlin *et al.*, 2018) qui prend comme entrée une séquence de sous-tokens, obtenue par la tokenisation WordPiece (Wu *et al.*, 2016) et produit une séquence de plongements contextuels de ces sous-tokens. Nous incorporons à la fois la question et le texte de référence dans l'entrée, comme illustré dans la figure 1. Les plongements d'entrée sont la somme des plongements de tokens et des plongements de segments. L'entrée est traitée de la manière suivante : intégration de mots clés (un mot clé [CLS] est ajouté aux mots d'entrée au début de la question et un mot clé [SEP] est inséré à la fin de la question et du texte de référence) et intégration de segments (un marqueur indiquant la question ou le texte de référence est ajouté à chaque mot clé).

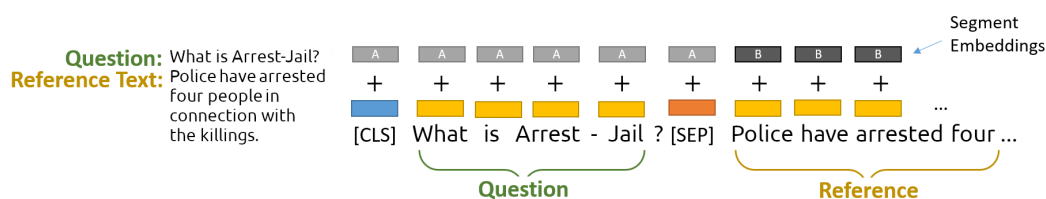


FIGURE 1 – Exemple de modification d'entrée pour s'adapter au paradigme QA pour une phrase contenant un événement de type *Attack*. La question est séparée par le mot clé [SEP] du texte de référence qui contient le déclencheur d'événement *war*.

Ensuite, pour chaque type d'événement [**Type d'événement**] (*Demonstration, Die, Attack, etc.*), nous formulons la question en les générant automatiquement à l'aide du modèle suivant : **What is [Type d'événement]?** La phrase "*Police have arrested four people in connection with the killings.*" contenant un événement *Arrest-Jail* est illustré dans la figure 1. Nous considérons également les questions qui n'ont pas de réponse dans le cas où un événement d'un type spécifique n'est pas présent dans la phrase. Lorsqu'il y a plus d'un événement du même type dans une phrase, nous considérons que la question a plusieurs réponses. Parmi les n meilleures réponses prédites, nous considérons toutes celles qui ont obtenu une probabilité supérieure à un seuil choisi (établi sur l'ensemble de développement). Lorsque les chunks prédits sont autonomes, nous ne considérons que le premier déclencheur d'événement prédit. Par exemple, si les morceaux nominaux *arrested* et *have arrested* sont prédits, mais seulement *arrested* est considéré. Ensuite, pour ajouter des informations sur les entités, nous augmentons les données d'entrée avec une série de tokens spéciaux. Ainsi, si nous considérons une phrase $x = [x_0, x_1, \dots, x_n]$ avec n tokens, nous augmentons x avec deux morceaux de mots réservés pour marquer le début et la fin de chaque entité d'événement ou mention d'argument dans la phrase.

Ensuite, nous proposons trois types de marqueurs : (1) *Marqueurs de position d'entité*, par exemple $\langle E \rangle$ et $\langle /E \rangle$ où E représente une entité de tout type, (2) *Marqueurs de type d'entité*, par exemple $\langle \text{PER} \rangle$ et $\langle /\text{PER} \rangle$ où PER représente une entité de type Personne, et (3) si les rôles d'argument

d'événement sont connus à l'avance, *Marqueurs de rôle d'argument*. Sur l'exemple précédent, killings est un déclencheur d'un événement *Die*, et arrested est un déclencheur d'un événement *Arrest-Jail*, *Police* est l'un des participants, une personne (*PER*) avec le rôle d'argument d'un *Agent*, et *four people* est également une entité personne (*PER*) avec le rôle d'argument *Person*. Les phrases modifiées avec les trois types de marqueurs sont : (1) “<E> **Police** </E> have arrested <E> **four people** </E> in connection with the killings.”; (2) “<PER> **Police** </PER> have arrested </PER> **four people** </PER> in connection with the killings.”; (3) “<Agent> **Police** </Agent> have arrested <Person> **four people** </Person> in connection with the killings.”.

En outre, un système de détection d'événements devrait détecter, dans la phrase présentée, le mot déclencheur killings pour un événement de type *Die* (cet événement a deux arguments *Police* et *quatre personnes*) et arrested pour un événement de type *Arrest-Jail* (cet événement n'a pas d'arguments). Pour les *Marqueurs de rôle d'argument*, si une entité a différents rôles dans différents événements présents dans la même phrase, nous marquons l'entité avec tous les rôles d'argument qu'elle possède.

3 Expériences

L'évaluation est réalisée sur le corpus ACE 2005 fourni par le programme ACE² et elle est basée sur les métriques standard : Précision (P), Rappel (R), et F-mesure (F1). À des fins de comparaison, nous utilisons le même ensemble de test avec 40 articles de presse (672 phrases), le même ensemble de développement avec 30 autres documents (863 phrases) et le même ensemble d'apprentissage avec les 529 documents restants (14 849 phrases) que dans les études précédentes (Ji *et al.*, 2008; Liao & Grishman, 2010). Le corpus ACE 2005 comporte 8 types d'événements, avec 33 sous-types (par exemple, le type d'événement *Conflict* a deux sous-types *Attack*, *Demonstrate*).

Approches	P	R	F1
MaxEnt avec des caractéristiques locales [♣] (Li <i>et al.</i> , 2013)	74.5	59.1	65.9
Cross-entity [♣] (Hong <i>et al.</i> , 2011)	72.9	64.3	68.3
DMCNN [♣] (Chen <i>et al.</i> , 2015)	75.6	63.6	69.1
Word CNN [♣] (Nguyen & Grishman, 2015)	71.8	66.4	69.0
Joint RNN [♣] (Nguyen <i>et al.</i> , 2016a)	66.0	73.0	69.3
BERT-QA	68.4	70.5	69.5
BERT-base (Du & Cardie, 2020)	67.1	73.2	70.0
Non-Consecutive CNN [♣] (Nguyen <i>et al.</i> , 2016b)	–	–	71.3
Attention-based ^{♣♥} (Liu <i>et al.</i> , 2017b)	78.0	66.3	71.7
BERT_QA_Trigger (Du & Cardie, 2020)	71.1	73.7	72.3
Graph CNN [♣] (Nguyen & Grishman, 2018)	77.9	68.8	73.1
BERT-QA + Marqueurs de position d'entité [♣]	78.0	70.7	74.2*
RCEE_ER [♣] (Liu <i>et al.</i> , 2020)	75.6	74.2	74.9
BERT-QA + Marqueurs de type d'entité [♣]	78.5	77.2	77.8*
BERT-QA + Marqueurs de rôle d'argument [♥]	83.2	80.5	81.8*

TABLE 1 – Évaluation de nos modèles et comparaison avec l'état de l'art pour la détection d'événement sur les données de test en aveugle. Les modèles avec [♣] utilisaient des mentions d'entité de référence. Les modèles avec [♥] utilisaient des arguments de référence. La signification statistique est mesurée avec le test de McNemar. * dénote une amélioration significative à $p \leq 0,01$.

2. <https://catalog.ldc.upenn.edu/LDC2006T06>

Évaluation générale Dans le tableau 1, nous présentons la comparaison entre notre modèle et les approches de l'état de l'art qui utilisent des informations sur les entités ou les arguments. Nous comparons avec le modèle basé sur MaxEnt avec des caractéristiques locales dans (Li *et al.*, 2013), la méthode basée sur les caractéristiques inter-entités extraite des événements en utilisant l'inférence inter-entités de la vérité de terrain (Hong *et al.*, 2011) et les modèles proposés par (Chen *et al.*, 2015), (Nguyen & Grishman, 2015), (Nguyen *et al.*, 2016b), et le cadre conjoint avec les RNN bidirectionnels (Nguyen *et al.*, 2016a) qui a expérimenté l'intégration des entités dans les modèles de détection d'évènement (DE) basés sur une architecture CNN. Nous comparons également la méthode proposée par (Liu *et al.*, 2017b) qui exploite également les informations sur les entités de manière explicite pour la DE via des mécanismes d'attention supervisée, et le CNN de graphes de (Nguyen & Grishman, 2018) qui a étudié un CNN basé sur des arbres de dépendance pour la DE avec une méthode de pooling qui s'appuie sur les mentions d'entités pour agréger les vecteurs de convolution. Nous comparons également avec les modèles où la tâche a été abordée comme une tâche d'QA, mais toujours formulée comme un problème de classification séquentielle visant à localiser les candidats au déclenchement, la ligne de base fine BERT-base-uncased et le BERT_QA_Trigger (Du & Cardie, 2020), et le RCEE_ER (Reading Compréhension pour Event Extraction, avec ER qui indique que le modèle a été finement ajusté sur les annotations des entités) (Liu *et al.*, 2020). Par rapport aux modèles de pointe précédents qui incluaient des informations sur les entités, nos modèles qui utilisent soit les positions, soit les types des entités, améliorent considérablement les performances de la détection des déclencheurs. Il est clair qu'un marquage supplémentaire des entités avec leurs types peut augmenter à la fois la précision et le rappel, équilibrant ainsi les scores finaux. Il convient de noter que, alors que les entités peuvent être présentes dans l'ensemble du document, les arguments ne peuvent entourer que les déclencheurs d'événements. Le fait de connaître à l'avance les rôles des arguments apporte des améliorations supplémentaires, nous supposons qu'une raison importante est que, puisque les arguments ne sont présents qu'autour des déclencheurs d'événements, cela pourrait aider le modèle de langage à être plus conscient de l'existence d'un événement ou de plusieurs événements dans une phrase.

Ambiguïté des déclencheurs Ainsi, comme appliqué par Madsen (2019), pour avoir une meilleure idée de la façon dont chaque modèle mémorise et utilise la mémoire pour la compréhension contextuelle, nous analysons la connectivité entre la sortie désirée et l'entrée. Celle-ci est calculée comme suit :

$$\text{connectivity}(t, \tilde{t}) = \left\| \frac{\partial y_k^{\tilde{t}}}{\partial x^t} \right\|_2$$

où t est l'indice de temps, \tilde{t} l'indice de temps de sortie, et le résultat est la magnitude du gradient entre les logits pour la sortie souhaitée $y_k^{\tilde{t}}$ et l'entrée x^t . La connectivité est calculée par rapport à la position de début et de fin de la réponse, puis elle est normalisée, et elle est visible sous forme de cartes de saillance pour chaque mot dans la figure 2.

En examinant les gradients de la figure 3, où deux événements de types différents sont présents, nous pouvons observer, dans la partie supérieure de la figure, que si le modèle considère les mots killings et arrested comme ayant un impact, il considère également les mots *police*, *connection* comme ayant un impact et sélectionne une réponse dans ce voisinage. Même si les deux candidats déclencheurs killings et arrested ont un impact évident en raison de leurs valeurs de gradient, en examinant les valeurs de probabilité, killings est reconnu avec une probabilité de 99,4%, tandis que arrested a obtenu une probabilité de $2,3 \times 10^{-7}$, valeur inférieure au seuil de 0,2 que nous avons choisi. Dans la partie

inférieure de la figure, pour la question *Qu'est-ce que Arrest-Jail ?*, les mots *die*, *police*, killings influencent clairement le choix de la réponse arrested.

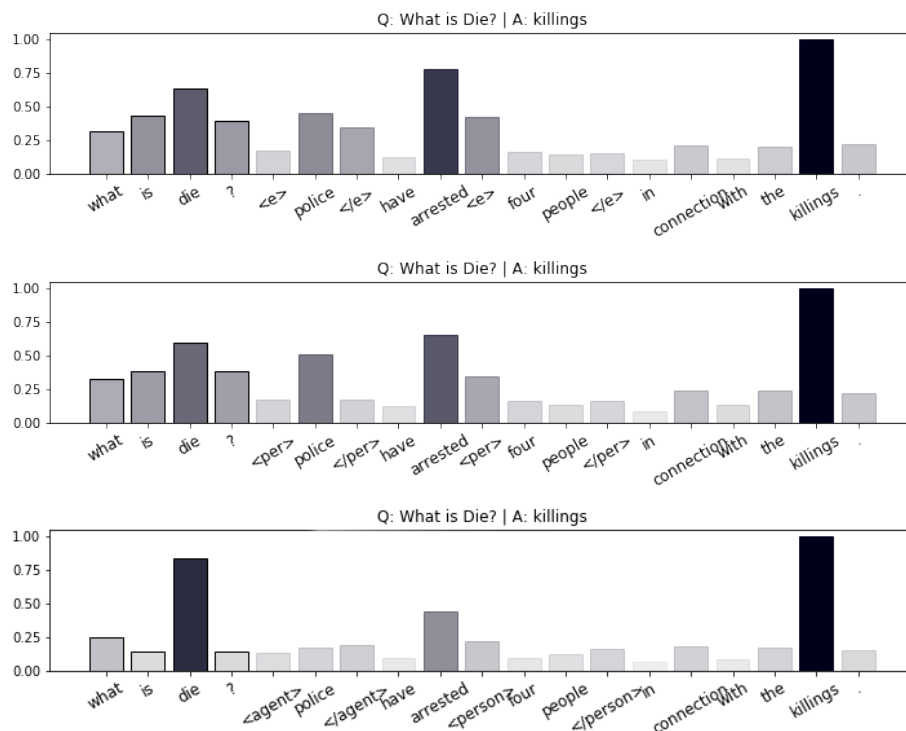


FIGURE 2 – Un exemple pour l’événement *Die* déclenché par killings avec les trois types de marqueurs : *Marqueurs de position d’entité*, *Marqueurs de type d’entité*, et *Marqueurs de rôle d’argument*.

En examinant les gradients de la figure 2, où deux événements de types différents sont présents, sans marqueurs et avec les trois types de modifications d’entrée : *Marqueurs de position d’entité*, *Marqueurs de type d’entité*, et *Marqueurs de rôle d’argument*, avec la question *What is Die ?* et la réponse correcte killings. Dans la partie supérieure de la figure, où la phrase a été augmentée avec les marqueurs de position d’entité, on remarque que les mots qui ont le plus d’impact dans le résultat sont killings ainsi que *die*, arrested et *police*. Dans ce cas, on peut également voir que le marqueur de fin </E> a également contribué. Dans la partie centrale de l’image, où la phrase a été augmentée des marqueurs de position d’entité, l’influence d’autres mots comme dans *die*, arrested, et *police* a légèrement diminué. Dans la partie inférieure de l’image, les gradients de ces mots sont visiblement réduits. Lorsque la phrase est complétée par des rôles d’arguments, le bruit autour de la bonne réponse a sensiblement diminué, réduit par les marqueurs supplémentaires. Les mots restants qui ont le plus d’impact sont le mot *die* dans la question et la réponse correcte killings.

Approches	P	R	F1
BERT-QA (pas entraîné sur ACE 2005)	0.7	8.3	1.3
BERT-QA	47.7	26.7	31.1
BERT-QA + Marqueurs de position d’entité♣	44.0	47.5	37.3
BERT-QA + Marqueurs de type d’entité♣	53.6	54.4	50.4
BERT-QA + Marqueurs de rôle d’argument♡	83.3	47.4	53.6

TABLE 2 – Évaluation de nos modèles sur des types d’événements inédits.

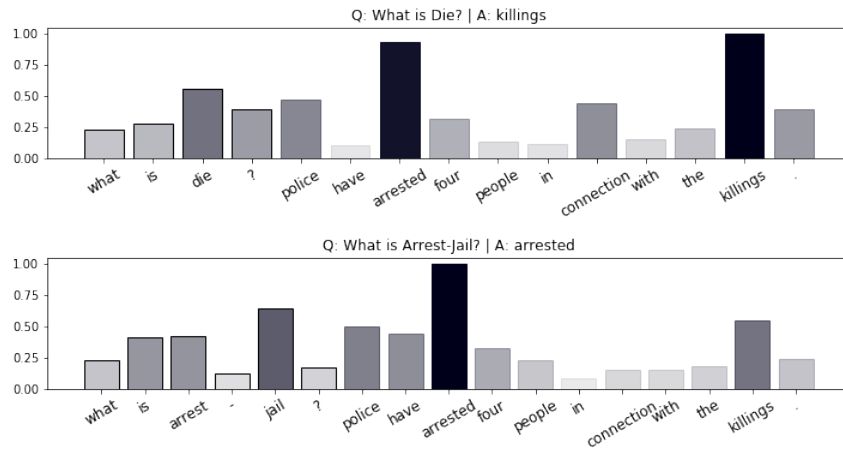


FIGURE 3 – Exemple d’une phrase qui contient deux événements : *Die* événement déclenché par le mot killings et *Arrest-Jail* événement déclenché par arrested. Le modèle utilisé est BERT-QA-base-uncased.



FIGURE 4 – Représentation du mot clé [CLS] de chaque phrase de l’ensemble de test contenant au moins un événement pour BERT-QA, BERT-QA + *Marqueurs de position d’entité*, BERT-QA-base-uncased + *Marqueurs de type d’entité* et BERT-QA + *Marqueurs de rôle d’argument*.

Qualité de la représentation des événements Afin d’analyser la qualité des représentations des phrases, nous extrayons la représentation [CLS] de chaque phrase pour BERT-QA et pour BERT-QA + *Marqueurs de rôle d’argument*. Ensuite, nous traçons ces représentations dans un espace à deux dimensions où les étiquettes (couleurs des points) sont les types d’événements, comme illustré dans la figure 4. Sur la représentation la plus à droite, où les *marqueurs de rôle d’argument* sont utilisés, il est clair que les groupes de représentations de phrases sont plus cohésifs que lorsqu’aucune information sur les entités n’est prise en compte (côté gauche), confirmant ainsi notre hypothèse concernant l’importance des caractéristiques informatives sur les entités dans un système de QA.

Capacité à détecter des événements non vus Dans le premier scénario, nous suivons la même stratégie que Du & Cardie (2020) où nous gardons 80% des types d’événements (27) dans l’ensemble d’apprentissage et 20% (6) des types d’événements non vus dans l’ensemble de test. Plus précisément, les types d’événements inconnus ont été choisis aléatoirement et sont les suivants : *Marriage*, *Trial-Hearing*, *Arrest-Jail*, *Acquit*, *Attack*, et *Declare-Bankruptcy*. Le tableau 2 présente les scores de performance de nos modèles pour les types d’événements non vus. Comme la majorité des modèles de la table 1 sont basés sur la classification de manière séquentielle, ils ne sont pas capables de traiter

des types d'événements non vus, et nous n'avons donc pas pu obtenir de valeurs de performance. D'après les résultats, sans aucune annotation d'événement, le BERT-QA obtient une faible valeur F1 (1.38%). Aussi, nous observons que les valeurs de performance augmentent proportionnellement à la spécificité des marqueurs.

Setting	N	Seen Event Types
A	1	Attack
B	3	Attack, Transport, Die
C	5	Attack, Transport, Die, Meet, Arrest-Jail
D	10	Attack, Transport, Die, Meet, Sentence, Arrest-Jail, Transfer-Money, Elect, Transfer-Ownership, End-Position

TABLE 3 – Types vus dans chaque paramètre de test tel que proposé par Huang *et al.* (2018).

Approches	Settings			
	A	B	C	D
Huang <i>et al.</i> (2018)	3.9	7.0	20.0	33.4
BERT-QA + Marqueurs de position d'entité♣	2.3	4.9	18.8	21.7
BERT-QA + Marqueurs de type d'entité♣	2.3	8.8	21.8	25.8
BERT-QA + Marqueurs de rôle d'argument♡	2.4	10.0	26.2	32.0

TABLE 4 – Évaluation de nos modèles sur des types d'événements invisibles (Hit@1 comme dans Huang *et al.* (2018)).

Dans un deuxième scénario, nous considérons une plus grande quantité d'événements non vus, et nous suivons la stratégie proposée par Huang *et al.* (2018), où sur le nombre total de types d'événements (33), nous sélectionnons les N types d'événements les plus populaires comme vus, et les autres comme non vus. N est fixé à 1, 3, 5 et 10 respectivement. Nous réalisons des expériences dans quatre configurations (A, B, C et D). Le tableau 3 montre les types qui ont été sélectionnés pour la formation dans chaque paramètre d'expérience. Le tableau 4 présente les scores de performance de nos modèles pour les types d'événements non vus. Nous nous attachons à montrer l'efficacité de nos méthodes en les comparant aux résultats de (Huang *et al.*, 2018). Nous observons d'abord, pour chaque modèle, que les valeurs de performance s'améliorent avec le nombre de type d'événements vus. Ensuite, on peut également remarquer que les scores augmentent proportionnellement à la spécificité des marqueurs.

4 Conclusions et perspectives

Nous concluons que les caractéristiques informatives supplémentaires apportées par la présence des entités et des rôles d'arguments dans le même contexte que les événements ont considérablement augmenté les performances du modèle, permettant d'atteindre des résultats à l'état de l'art. De plus, ce type de modèle qui utilise les informations sur les entités a permis de tirer parti de l'ambiguïté des déclencheurs d'événements et ainsi démontré son potentiel dans la détection de types d'événements non vus.

Remerciements

Ce travail a été soutenu par les projets ANNA et Termitrad financés par la Région Nouvelle-Aquitaine.

Références

- BOROS E., MORENO J. G. & DOUCET A. (2021). Event detection with entity markers. In D. HIEMSTRA, M. MOENS, J. MOTHE, R. PEREGO, M. POTTHAST & F. SEBASTIANI, Édts., *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part II*, volume 12657 de *Lecture Notes in Computer Science*, p. 233–240 : Springer. DOI : [10.1007/978-3-030-72240-1_20](https://doi.org/10.1007/978-3-030-72240-1_20).
- CHEN Y., XU L., LIU K., ZENG D. & ZHAO J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 167–176.
- DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.
- 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.
- HONG Y., ZHANG J., MA B., YAO J., ZHOU G. & ZHU Q. (2011). Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies-Volume 1*, p. 1127–1136 : Association for Computational Linguistics.
- 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).
- JI H., GRISHMAN R. *et al.* (2008). Refining event extraction through cross-document inference. In *ACL*, p. 254–262.
- LI Q., JI H. & HUANG L. (2013). Joint event extraction via structured prediction with global features. In *ACL (1)*, p. 73–82.
- LI W., CHENG D., HE L., WANG Y. & JIN X. (2019). Joint event extraction based on hierarchical event schemas from framenet. *IEEE Access*, **7**, 25001–25015.
- LIAO S. & GRISHMAN R. (2010). Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, p. 789–797 : Association for Computational Linguistics.
- LIU J., CHEN Y., LIU K., BI W. & LIU X. (2020). Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1641–1651.
- LIU S., CHEN Y., LIU K. & ZHAO J. (2017a). Exploiting argument information to improve event detection via supervised attention mechanisms. In *55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, p. 1789–1798, Vancouver, Canada.
- LIU S., CHEN Y., LIU K. & ZHAO J. (2017b). Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1789–1798.
- LIU X., LUO Z. & HUANG H. (2018). Jointly multiple events extraction via attention-based graph information aggregation. *arXiv preprint arXiv :1809.09078*.

- MADSEN A. (2019). Visualizing memorization in rnns. *Distill*. <https://distill.pub/2019/memorization-in-rnns>, DOI : [10.23915/distill.00016](https://doi.org/10.23915/distill.00016).
- MORENO J. G., DOUCET A. & GRAU B. (2021). Relation classification via relation validation. In *Proceedings of the 6th Workshop on Semantic Deep Learning (SemDeep-6)*, p. 20–27.
- NGUYEN T. H., CHO K. & GRISHMAN R. (2016a). Joint event extraction via recurrent neural networks. In *Proceedings of NAACL-HLT*, p. 300–309.
- NGUYEN T. H., FU L., CHO K. & GRISHMAN R. (2016b). A two-stage approach for extending event detection to new types via neural networks. *ACL 2016*, p. 158.
- NGUYEN T. H. & GRISHMAN R. (2015). Event detection and domain adaptation with convolutional neural networks. In *ACL (2)*, p. 365–371.
- NGUYEN T. H. & GRISHMAN R. (2018). Graph convolutional networks with argument-aware pooling for event detection. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018)*.
- WU Y., SCHUSTER M., CHEN Z., LE Q. V., NOROUZI M., MACHEREY W., KRICKUN M., CAO Y., GAO Q., MACHEREY K. *et al.* (2016). Google’s neural machine translation system : Bridging the gap between human and machine translation. *arXiv preprint arXiv :1609.08144*.