Intégration de contexte global par amorçage pour la détection d'événements

Dorian Kodelja Romaric Besançon Olivier Ferret
CEA, LIST, Laboratoire Vision et Ingénierie des Contenus, Gif-sur-Yvette, F91191 France
dorian.kodelja, romaric.besancon, olivier.ferret@cea.fr

RÉSUMÉ

Les approches neuronales obtiennent depuis plusieurs années des résultats intéressants en extraction d'événements. Cependant, les approches développées dans ce cadre se limitent généralement à un contexte phrastique. Or, si certains types d'événements sont aisément identifiables à ce niveau, l'exploitation d'indices présents dans d'autres phrases est parfois nécessaire pour permettre de désambiguïser des événements. Dans cet article, nous proposons ainsi l'intégration d'une représentation d'un contexte plus large pour améliorer l'apprentissage d'un réseau convolutif. Cette représentation est obtenue par amorçage en exploitant les résultats d'un premier modèle convolutif opérant au niveau phrastique. Dans le cadre d'une évaluation réalisée sur les données de la campagne TAC 2017, nous montrons que ce modèle global obtient un gain significatif par rapport au modèle local, ces deux modèles étant eux-mêmes compétitifs par rapport aux résultats de TAC 2017. Nous étudions également en détail le gain de performance de notre nouveau modèle au travers de plusieurs expériences complémentaires.

ABSTRACT

Integrating global context via bootstrapping for event detection.

Over the last few years, neural models developed for event extraction have reached an interesting level of results. However, their application is generally limited to sentences, which is sufficient for identifying certain types of events but too limited in terms of scope for disambiguating some occurrences of events. In this article, we propose to integrate in a convolutional neural network the representation of contexts beyond the level of sentences. This representation is built following a bootstrapping approach by exploiting an intra-sentential convolutional model. Within the evaluation framework of TAC 2017, we show that our global model significantlty outperforms the intra-sentential model while the two models are competitive with the results obtained by TAC 2017 participants. We furthermore analyze the gain of our proposed model through supplementary experiments.

MOTS-CLÉS : Détection d'événement, réseau de neurones convolutifs, contexte discursif.

KEYWORDS: Event detection, convolutional neural networks, discourse context.

1 Introduction

L'extraction d'événements supervisée consiste à identifier au sein d'un document les occurrences de types d'événements préalablement définis. Ces événements sont caractérisés par des interactions entre plusieurs entités y tenant des rôles spécifiques. Cette tâche est le plus souvent décomposée en plusieurs tâches de classification successives pour identifier d'abord la mention d'un événement puis ses arguments. Nous nous intéressons ici à la première étape, aussi appelée détection d'événements ou détection de mentions événementielles, tâche qui consiste à identifier dans le texte le ou les mots indiquant le plus clairement la présence d'un événement.

Les approches actuelles sont majoritairement fondées sur des modèles neuronaux, qu'il s'agisse de modèles convolutifs (Chen *et al.*, 2015; Nguyen & Grishman, 2015), récurrents (Nguyen *et al.*, 2016a) ou combinant les deux approches (Feng *et al.*, 2016). Le problème est alors modélisé sous forme d'une tâche de classification pour chaque mot du document. Les meilleurs systèmes des campagnes d'évaluation récentes sur le sujet relèvent par ailleurs tous de ce même paradigme.

Le jeu de données que nous utilisons par la suite est annoté selon la taxonomie Rich ERE (Song *et al.*, 2015). Ce schéma d'annotation distingue 38 sous-types d'événements répartis en 9 types. Ces types d'événements couvrent un large champ d'interactions possibles, allant des transactions financières aux conflits armés en passant par les correspondances écrites ou les licenciements. L'étendue et la finesse de cette modélisation amènent à distinguer plusieurs sources d'erreurs de détection des événements. L'une d'elles réside dans l'ambiguïté des marqueurs. Par exemple, la polysémie d'un verbe peut être source de confusion entre plusieurs types d'événements. Le mot "*fired*" peut ainsi faire référence à un coup de feu, indicateur d'un événement de type "*Conflict-Attack*", ou signifier "licencier" et indiquer un événement de type "*Personnel-End-Position*". Sur un autre plan, la proximité des sous-types d'événements appartenant à un même type peut rendre ceux-ci distinguables seulement par une compréhension fine de leur contexte et constitue de ce fait une autre source d'erreurs possible. Ainsi, le type d'événement "*Contact*" distingue les sous-types "*Contact-Broadcast*" lorsque la communication est à sens unique, "*Contact-Meet*" pour une rencontre physique entre plusieurs personnes, "*Contact-Correspondance*" s'il s'agit d'un échange à distance et "*Contact-Contact*" quand aucun sous-type plus spécifique ne correspond.

On voit ici l'importance de la prise en compte du contexte pour résoudre les différentes ambiguïtés possibles entre différents types. Cependant, si les systèmes présentés précédemment parviennent à identifier correctement une grande partie des mentions d'événements, des ambiguïtés subsistent lorsque le contexte local n'est pas assez informatif pour permettre au modèle de discriminer convenablement un type d'événements d'un autre ou de l'absence d'événement. On peut ainsi distinguer deux types d'ambiguïtés en fonction de la portée du contexte nécessaire à leur résolution :

— « Le rappeur lyonnais **déclara** "Les instruments c'était mieux à vent" aux micros de France Inter. »
— « Les **départs** se multiplient chez l'opérateur téléphonique. [...] Plus de 200 démissions ont ainsi été reçues le mois dernier. »

Dans le premier exemple, l'ambiguïté fréquente entre les types d'événements *Contact-contact* et *Contact-broadcast* peut être résolue en identifiant que la communication s'adresse à un média et se fait donc à sens unique. Mais la distance entre *déclara* et *France Inter* peut rendre cette information difficile à exploiter bien qu'elle soit locale à la phrase. Les modèles, en particulier convolutifs, n'exploitent en pratique qu'un contexte très restreint autour de la mention candidate à étiqueter

pour prendre une décision, même lorsque le contexte fourni est plus large. C'est ici un problème de désambiguïsation intra-phrastique. Dans le second exemple, le contexte local de la phrase n'indique pas clairement que *départ* fait référence à un événement de type *End_Position*. Mais la thématique des licenciements est clairement identifiable plus loin dans le document. Il s'agit ici d'un problème de désambiguïsation inter-phrastique.

Pour réaliser ces désambiguïsations, il est nécessaire d'exploiter un contexte plus global. Duan *et al.* (2017) mettent en avant l'intérêt d'une telle exploitation afin de prendre en compte la cohérence interne des documents sur le plan thématique : un document traitant d'un conflit armé présentera plus d'événements de type *Die* ou *Attack* que de naissances. Ils proposent de fournir en entrée d'un BiLSTM une représentation distribuée du document apprise de manière non supervisée (Le & Mikolov, 2014). Ce contexte global n'est donc pas spécialisé pour la tâche cible et n'est en outre adapté qu'au problème de désambiguïsation inter-phrastique. Notre approche vise au contraire à apprendre une représentation des documents en lien avec la tâche cible et ce, pour les deux cas de figure identifiés précédemment. Nous utilisons pour ce faire une méthode d'amorçage, en définissant un premier modèle à un niveau local (niveau des mots) et en l'appliquant à l'ensemble du document. Les prédictions locales ainsi réalisées sont agrégées pour obtenir un vecteur de contexte pour chaque document. Ces vecteurs sont alors intégrés à un nouveau modèle exploitant ce contexte.

Dans la suite de cet article, nous présentons dans un premier temps nos modèles local et global à la section 2. Puis, dans la section 3.2, nous étudions l'influence de la taille du contexte local sur les performances et observons l'incapacité du modèle de base à exploiter l'information distante, démontrant ainsi l'intérêt de l'intégration d'un contexte distant. La section 3.3 compare les différents types de représentations du contexte global utilisables. Enfin, nous évaluons à la section 3.4 différentes modalités d'intégration de cette représentation globale au modèle local et les confrontons à plusieurs baselines sur les données de la campagne d'évaluation TAC Event Nugget 2017. Ces expériences montrent que ce nouveau modèle obtient des gains significatifs par rapport au modèle local *baseline* et s'avère compétitif par rapport à d'autres approches neuronales. En dernier lieu, nous étudions plus en détail l'apport de la représentation globale à la section 4.

2 Description de l'approche

Comme nous l'avons vu en introduction, la détection d'événements consiste à identifier dans un texte les mentions d'événements et leur associer un type selon une taxonomie préalablement établie. Dans cet article, nous nous appuyons sur les 38 types d'événements de la taxonomie DEFT Rich ERE utilisée dans le cadre des campagnes TAC. Dans les annotations liées à cette taxonomie, les mentions d'événements étant en grande majorité des mots simples (Reimers & Gurevych, 2017), nous choisissons d'aborder le problème non pas comme une tâche d'annotation de séquences mais comme une tâche de classification multi-classe de mots. Ce choix est d'un impact négatif négligeable mais simplifie la modélisation et permet l'introduction d'un vecteur de positions contribuant grandement aux performances (Nguyen *et al.*, 2016b). Enfin, dans la continuité des approches neuronales récentes, nous nous plaçons à l'échelle intra-phrastique. La tâche est alors envisagée comme un problème de classification multi-classe.

Nous présentons à la figure 1 la procédure générale d'intégration du contexte à un modèle convolutif de détection d'événements. Un premier modèle $\text{CNN}_{\text{local}}$ est entraîné pour associer des étiquettes d'événement à chaque mot d'un document. Ces étiquettes sont agrégées au niveau d'un contexte

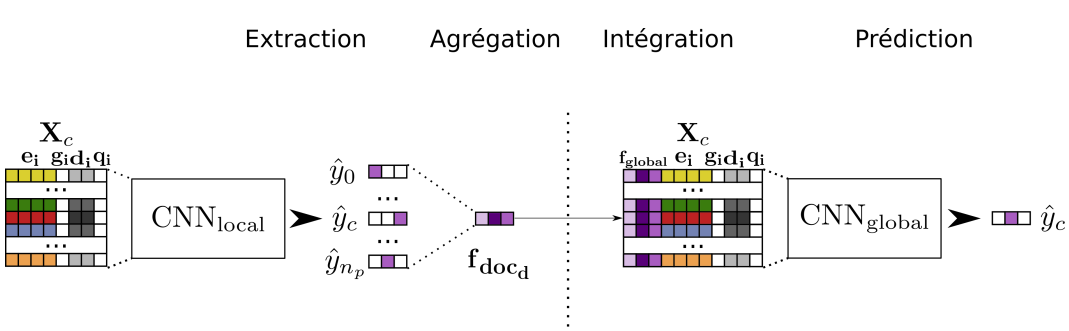

FIGURE 1 – Principe d'intégration du contexte par amorçage. $\hat{y}_c$ est la prédiction du modèle pour la mention courante, $\hat{y}_0$, la prédiction pour la première mention trouvée du document et $\hat{y}_{n_p}$ la dernière mention trouvée du document.

(dans la figure, ce contexte est le document) et ajoutées en entrée d'un nouveau modèle $\mathrm{CNN}_{\mathrm{global}}$. Nous présentons plus en détail ces modèles local et global dans les sections suivantes.

2.1 Modèle local de détection d'événements

Notre modèle de détection d'événements au niveau local s'appuie sur un réseau de neurones convolutif inspiré de l'architecture proposée par (Nguyen *et al.*, 2016b). Nous considérons successivement chaque mot de chaque phrase en tant que mention candidate. Cette mention est représentée par un contexte local de taille fixe centré sur ce mot. Si le contexte local dépasse les limites de la phrase courante, un token spécial est utilisé pour compléter la séquence. Soit i_c l'index de la mention candidate et w la taille de la fenêtre. On définit $\mathbf{t}_c = [i_{c-w}, i_{c-w+1}, \ldots, i_c, \ldots, i_{c+w-1}, i_{c+w}]$ le vecteur des index du contexte local centré sur i_c . Ce vecteur d'index est transformé en une matrice de réels $\mathbf{X}_c = [\mathbf{x_{c-w}}, \mathbf{x_{c-w+1}}, \ldots, \mathbf{x_c}, \ldots, \mathbf{x_{c+w-1}}, \mathbf{x_{c+w}}]$ en remplaçant chaque index i par une représentation $\mathbf{x_i} = [\mathbf{e_i}, \mathbf{d_i}, \mathbf{g_i}, \mathbf{q_i}]$ obtenue en combinant les différentes représentations suivantes :

Plongement de mot $\mathbf{e_i}$ Cette représentation distribuée du mot t_i est pré-entraînée sur un large corpus pour capter des informations sémantiques et syntaxiques à propos de ce mot (Mikolov *et al.*, 2013).

Vecteur de position $\mathbf{d_i}$ Ce vecteur encode la position relative i du mot t_i par rapport au candidat t_0.

Vecteur des dépendances syntaxiques $\mathbf{g_i}$ Ce vecteur a une dimension correspondant au nombre de dépendances considérées. Si une dépendance d'un certain type existe entre t_i et t_0, la dimension correspondante du vecteur est égale à 1. Dans nos expériences, nous utilisons les dépendances de base (*basic dependencies*) fournies par l'outil Stanford CoreNLP (Manning *et al.*, 2014).

Vecteur de syntagme $\mathbf{q_i}$ Ce vecteur encode le type de constituant syntaxique dont le token fait partie sous la forme d'une annotation IOB fournie par un *chunker* [1]. Cette représentation est construite à partir de l'arbre syntaxique fourni par l'outil Stanford CoreNLP.

À partir de cette matrice d'entrée $\mathbf{X}_c$, nous appliquons une couche de convolution constituée de plusieurs filtres de tailles différentes. Une couche de *global max-pooling* est ensuite appliquée afin d'obtenir une seule valeur pour chaque filtre. Nous obtenons ainsi une représentation du candidat dans son contexte local apprise par le réseau convolutif. Cette représentation locale $\mathbf{f_{softmax}} = [\mathbf{f_{pooling}}]$ est ensuite fournie en entrée d'une couche de neurones entièrement connectée (*fully connected*) dotée d'un softmax. Ce dernier étage permet de calculer la distribution de probabilités des différentes

1. https://github.com/mgormley/concrete-chunklink

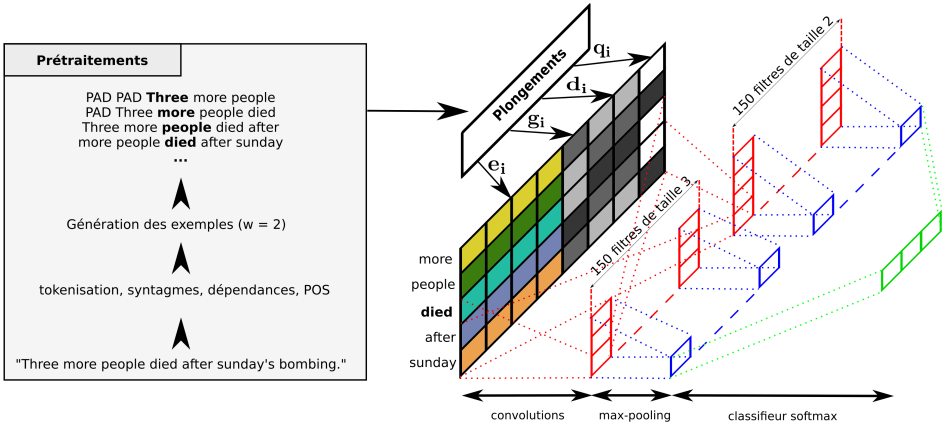

FIGURE 2 – Architecture du modèle $\text{CNN}_{\text{local}}$

classes d'événements pour le candidat et d'en déduire un label unique $\hat{y}_c$ en prenant la classe de probabilité maximale. Pour améliorer la généralisation, un dropout est appliqué sur la couche d'entrée du réseau. La figure 2 représente de manière détaillée l'architecture du $\text{CNN}_{\text{local}}$.

2.2 Intégration de contexte dans un modèle global

Afin d'augmenter les performances de notre modèle convolutif, nous proposons d'intégrer une information de contexte plus large, sous la forme d'une représentation globale focalisée sur la tâche d'extraction d'événement en utilisant un principe d'amorçage. Pour ce faire, nous réalisons un premier entraînement du modèle local présenté précédemment. Nous utilisons ensuite ce modèle pour extraire $\hat{y}_c$ pour chaque mot du corpus. Nous agrégeons alors $\hat{y}_c$ par *sum-pooling*, ce qui équivaut à construire un histogramme des différents types d'événements détectés. Nous réalisons cette agrégation à trois niveaux différents : à l'échelle de chaque phrase (*phrase*), d'un contexte de trois phrases centré sur la phrase courante (*large*) ou à l'échelle du document (*doc*).

Nous utilisons la notation suivante pour désigner les différentes configurations possibles de contexte global d'une phrase : $\mathbf{f_{global}} = \mathbf{f_{[doc/large/phrase]}}$. Le contexte global $\mathbf{f_{global}}$ peut être intégré au niveau de la matrice d'entrée $\mathbf{X}_c$ en redéfinissant $\mathbf{x_i} = [\mathbf{e_i}, \mathbf{d_i}, \mathbf{g_i}, \mathbf{q_i}, \mathbf{f_{global}}]$ ou concaténé avant la couche entièrement connectée : $\mathbf{f_{softmax}} = [\mathbf{f_{pooling}}, \mathbf{f_{global}}]$. On distinguera ainsi six modèles en fonction des niveaux d'agrégation et d'intégration avec la notation $\text{CNN}_{\text{[doc/large/phrase]-[plongement/softmax]}}$.

3 Expériences

3.1 Paramètres et ressources

Nous utilisons dans nos expériences les *plongements* à 300 dimensions pré-entraînés sur Google News avec *word2vec* en les modifiant durant l'entraînement. Les vecteurs de positions et de syntagmes sont de taille 50. La probabilité de *dropout* est fixée à 0,8. Pour chacune des tailles de champ récepteur (2,3,4,5), 150 filtres sont utilisés, pour un total de 600. Ces filtres sont dotés d'une tangente

hyperbolique comme non-linéarité. Le modèle est entraîné par descente de gradient stochastique (SGD) avec l'optimiseur Adagrad et un *clipping* du gradient fixé à 3. La taille des mini-lots est fixée à 50. Le nombre d'époques d'apprentissage est contrôlé par *early stopping* sur le jeu de validation. Les résultats présentés sont des moyennes sur 10 exécutions en utilisant le score micro-f1 de l'outil officiel d'évaluation de TAC 2017. Notre corpus d'entraînement est constitué de l'union des jeux de données DEFT_RICH_ ERE_R2_V2 (LDC2015E68), DEFT_RICH_ ERE_V2 (LDC2015E29) et TAC 2015 (LDC2017E02). Notre corpus de validation est le jeu de données issu de la campagne TAC 2016 (LDC2017E02) et nous nous testons sur les données de la campagne TAC 2017 Event Nugget (LDC2017E02). Il existe au sein de ces jeux de données quelques rares cas de mentions annotées avec plusieurs types d'événements distincts. Parmi ces cas de figure, la grande majorité appartient à l'une des trois combinaisons suivantes : (*Attack/Die*, *Transfer-Money/Transfer-Ownership*, *Attack/Injure*). Pour traiter ce problème, nous introduisons trois types hybrides lors de l'apprentissage pour ces trois types d'événements, ce qui permet de conserver une classification simple (mono-étiquette). Nos jeux de données de validation et de test se focalisent sur les types d'événements les plus difficiles de Rich ERE, à l'instar de TAC 2017, et restreignent la tâche à 19 des 38 types. Nous entraînons toutefois notre modèle sur l'ensemble des 42 classes (classes hybrides et classe nulle incluse) mais nous ignorons les types non présents en test lors de la prédiction. De même, le vecteur global n'agrège que les prédictions des types présents sur le jeu de test. Enfin, différentes normalisations du vecteur de contexte global ont été comparées expérimentalement. Les résultats présentés ici reposent sur la meilleure normalisation obtenue en validation pour chaque configuration : les vecteurs $\mathbf{f}_{\mathbf{[large/phrase]}}$ ne sont pas normalisés alors que le vecteur $\mathbf{f}_{\mathbf{doc}}$ est centré-réduit avant d'être fourni au modèle.

3.2 Influence de la taille du contexte local

Les modèles convolutifs obtenant des performances compétitives sur les bases ACE 2005 (Nguyen & Grishman, 2015), TAC 2016 (Nguyen *et al.*, 2016b) et TAC 2017 (Kodelja *et al.*, 2017) partagent des architectures similaires, notamment concernant la taille du contexte local employé. La fenêtre utilisée est de taille $w = 15$, centrée sur la mention candidate. Afin d'observer la capacité du modèle à exploiter des dépendances longues au sein de la fenêtre, nous présentons à la figure 3 les performances du modèle en fonction de la taille de ce contexte. On constate que les performances du modèle local saturent dès $w = 2$, taille que nous conservons par la suite. Ces résultats indiquent que le modèle convolutif ne parvient pas véritablement à exploiter des dépendances distantes au sein du contexte local. Ce modèle local obtient des performances similaires à celles d'un ensemble de réseaux BiLSTM (voir (Makarov & Clematide, 2017) dans le tableau 2). Il semble donc que cette architecture récurrente, bien que théoriquement mieux à même d'exploiter des dépendances longues, ne le fait en réalité pas mieux qu'un modèle convolutif. Ces résultats motivent l'intérêt théorique de notre approche : les réseaux ne parvenant pas à exploiter plus d'informations lorsque l'on augmente la taille de la fenêtre du contexte local, il est souhaitable de rendre cette information distante accessible. Les sections suivantes se consacrent à l'étude de cette intégration.

3.3 Influence de la taille du contexte global

Comme nous l'avons vu précédemment, l'agrégation du contexte peut se faire à plusieurs échelles. L'agrégation au niveau de la phrase courante peut résoudre les ambiguïtés intra-phrastiques alors qu'une agrégation plus large peut être utile pour la désambiguïsation inter-phrastique. Afin de déterminer le niveau d'agrégation le plus utile, nous comparons dans le tableau 1 l'apport de l'intégration

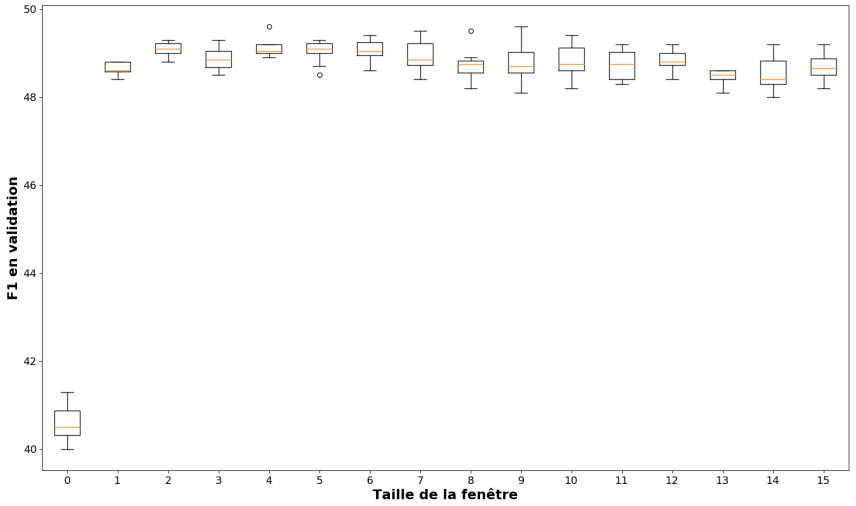

FIGURE 3 – Influence de la taille w du contexte local sur les performances du modèle local en validation. Pour chaque configuration, les résultats sont des moyennes sur 8 entraînements.

méthode	P	R	F
$\text{CNN}_{\text{doc-plongement}}$	**52,71**	47,95	**50,2** ‡
$\text{CNN}_{\text{large-plongement}}$	52	47,6	49,69
$\text{CNN}_{\text{phrase-plongement}}$	49,83	49,49	49,66
$\text{CNN}_{\text{local}}$	46,42	**52,04**	49,06

TABLE 1 – Performances sur la base de validation TAC 2016 en fonction du niveau d'agrégation. Résultats moyennés sur 10 entraînements pour chaque configuration. Seul le modèle $\text{CNN}_{\text{doc-plongement}}$ est significativement meilleur que $\text{CNN}_{\text{local}}$ ($p < 0,01$).

du contexte $\mathbf{f}_{\mathbf{global}}$ à la matrice d'entrée $\mathbf{X}_c$ en fonction du niveau d'agrégation : *phrase*, *large* et *doc*. Nous constatons tout d'abord que les trois niveaux d'agrégation améliorent significativement les performances par rapport à notre baseline $\text{CNN}_{\text{local}}$. De plus, les performances augmentent avec la taille du contexte.

3.4 Comparaison avec l'état de l'art

Afin de valider l'apport de notre méthode, en plus de la comparaison à notre modèle initial $\text{CNN}_{\text{local}}$, nous nous comparons aux 3 modèles ayant obtenus les meilleurs résultats lors de la campagne d'évaluation TAC 2017 :

1. **Méthode d'ensemble BiLSTM CRF** : Jiang *et al.* (2017) utilisent un ensemble de 10 modèles BiLSTM combinés par une stratégie de vote. Mettant en avant le bon rappel des modèles neuronaux au détriment de la précision, ils y adjoignent un classifieur CRF pour améliorer la précision. Pour le BiLSTM, seuls des plongements de mots sont employés, tandis que le CRF

Méthodes	max			moyenne sur 10 exécutions		
	P	R	F	P	R	F
BILSTM CRF (Jiang) †	**56,83**	**55,57**	**56,19**	-	-	-
BILSTM à large marge (Makarov) †	52,16	48,71	50,37	-	-	-
CNN (Kodelja)	54,23	46,59	50,14	-	-	-
CNN_{local}	52,21	49,55	50,84	51,9	48,92	50,36
$CNN_{doc\text{-}plongement}$	**59,13**	45,37	51,34	**58,07**	45,43	50,95 ‡
$CNN_{doc\text{-}softmax}$	52,87	**50,35**	**51,58**	53,12	**49,61**	**51,3** ‡
$CNN_{doc\text{-}plong\_soft}$	55,72	47,08	51,04	57,62	45,09	50,58
$CNN_{doc2vec}$	53,20	47,40	50.10	53,54	46,92	49,98

TABLE 2 – Performance sur la base de test TAC 2017. "†" désigne des modèles d'ensemble. ‡ indique dans la seconde partie du tableau les modèles significativement meilleurs que le modèle CNN_{local} ($p < 0,01$ pour un t-test bilatéral sur les moyennes).

emploie de multiples attributs tels que tokens, lemmes, racines, présence d'entités nommées et étiquettes morphosyntaxiques.

2. **BiLSTM à large marge** : Makarov & Clematide (2017) utilisent un BiLSTM doté d'un objectif à large marge (Gimpel & Smith, 2010). Cet objectif pénalise plus fortement les faux négatifs afin de compenser la rareté des classes positives dans le jeu de données. Un ensemble de 5 réseaux est utilisé pour la prédiction et des types hybrides sont utilisés.
3. **Modèle convolutif** : ce modèle, proposé par (Kodelja *et al.*, 2017) est similaire à notre modèle CNN_{local}. Il s'agit d'un réseau convolutif utilisant des plongements de mots, de positions, de parties du discours et des dépendances syntaxiques en entrée du modèle. La principale différence est l'absence de types hybrides pour gérer les cooccurrences d'événements.
4. **CNN-doc2vec** : à l'instar de Duan *et al.* (2017), nous intégrons un vecteur de document au niveau des plongements de notre modèle. Ce vecteur de taille 100 est généré par le modèle PV-DM (Le & Mikolov, 2014). À la différence de notre représentation globale, celle-ci n'est pas spécifique à la tâche. Nous avons optimisé les mêmes hyperparamètres d'intégration que pour notre représentation, à savoir le choix de la normalisation et du niveau d'intégration. La meilleure configuration présentée ici intègre des vecteurs centrés-réduits au niveau du softmax.

Le tableau 2 présente la comparaison de ces méthodes sur la base de test TAC 2017. Il est difficile de comparer notre contribution à (Jiang *et al.*, 2017) et (Makarov & Clematide, 2017) car ce sont des méthodes d'ensemble alors que nous présentons les performances pour un seul modèle. De plus, leurs scores moyens sur plusieurs initialisations ne sont pas disponibles alors que les variations sont souvent non négligeables (Reimers & Gurevych, 2017).

Le modèle hybride de Jiang *et al.* (2017) n'est en outre pas une méthode d'ensemble simple fondée sur le vote de plusieurs modèles de même architecture mais la combinaison d'un ensemble de BiLSTMs votant pour une prédiction agrégée avec la prédiction d'un CRF selon une heuristique spécifique. Il est à noter de ce point de vue que notre approche ne faisant pas d'hypothèse sur le modèle neuronal de base, il serait théoriquement possible d'intégrer notre représentation globale aux BiLSTMs avant l'application de la stratégie d'ensemble.

Un autre élément rendant les comparaisons difficiles, identifié lors d'analyses récentes, est la pré-

sence de blocs de citations dans les documents. Ces citations ne sont pas annotées en événements, même lorsqu'elles reprennent des phrases précédentes contenant effectivement des événements. Ces phrases constituent donc des duplicatas de phrases contenant possiblement des événements. Durant l'apprentissage, le modèle reçoit alors des annotations contradictoires pour deux exemples pourtant identiques. Durant le test, ignorer ces phrases peut ainsi significativement augmenter les performances. Les résultats présentés ici ne tiennent pas compte de cet aspect, ce qui minore très certainement les performances de notre modèle. Nous avons eu par ailleurs confirmation [2] que les performances rapportées par Makarov & Clematide (2017) négligeaient ce phénomène et que sa prise en compte dans leur modèle entraînait également un gain significatif.

Leur approche peut donc être comparée à la nôtre de ce point de vue [3], comparaison montrant que notre *baseline* $\text{CNN}_{\text{local}}$ est légèrement supérieure en moyenne aux performances du BiLSTM à large marge de Makarov & Clematide (2017). Malgré la tendance actuelle à privilégier les architectures récurrentes, les modèles utilisant la convolution restent donc compétitifs. On peut supposer que de façon similaire à ce que nous avions constaté à la section 3.2 pour les CNNs, les RNNs utilisés n'apprennent pas à exploiter réellement l'intégralité du contexte à disposition, l'influence du contexte proche étant prédominante dans la majorité des cas. Enfin, pour achever la comparaison avec les modèles extérieurs, le tableau 2 montre que l'introduction de types hybrides dans notre baseline $\text{CNN}_{\text{local}}$ permet d'obtenir de meilleurs résultats que ceux du CNN de Kodelja *et al.* (2017).

Concernant plus spécifiquement les modèles proposés, les variantes $\text{CNN}_{\text{doc-plongement}}$ et $\text{CNN}_{\text{doc-softmax}}$ améliorent de manière significative les performances par rapport à notre *baseline* et au modèle d'ensemble de Makarov & Clematide (2017). L'intégration simultanée aux deux niveaux, $\text{CNN}_{\text{doc-plong\_soft}}$, n'obtient pas en revanche de gain significatif. Enfin, on peut observer que l'intégration de la représentation globale proposée par (Duan *et al.*, 2017) provoque une chute des performances. L'absence de spécificité des représentations construites par rapport à la tâche et au corpus considérés est une explication possible de cette contre-performance.

4 Discussions

4.1 Analyse du choix du contexte

En premier lieu, il faut souligner que du point de vue de la taille du contexte à prendre en compte, les résultats de la section 3.2 montrent assez clairement l'intérêt de se situer à l'échelle du document plutôt qu'à une granularité de contexte plus fine. Une interprétation possible de ce constat est que l'intégration d'un contexte global au modèle est intrinsèquement plus adaptée à la résolution des ambiguïtés inter-phrastiques qu'intra-phrastiques. Sur un autre plan, on peut également noter que le contexte global est généré à partir des prédictions d'un premier modèle imparfait. Il est donc bruité. Agréger les prédictions sur un contexte plus large pourrait alors permettre de compenser ce bruit plus efficacement. Afin de distinguer ces deux phénomènes, nous comparons dans le tableau 3 les résultats de l'intégration au niveau de la phrase et du document en utilisant cette fois les annotations réelles à la place des prédictions du $\text{CNN}_{\text{local}}$. On constate que cette fois encore, les meilleures performances sont obtenues en agrégeant l'information à l'échelle du document. L'écart avec $\text{CNN}_{\text{phrase-plongement}}$ est même encore plus élevé. Il semble donc que le niveau d'agrégation ne dépende pas des performances

2. Communication personnelle.
3. Ce que nous ne pouvons pas dire en revanche concernant (Jiang *et al.*, 2017).

méthode	P	R	F
$CNN_{\text{doc-plongement}}$	**54,85**	51,02	**52,83**
$CNN_{\text{phrase-plongement}}$	54,21	47,58	50,68
CNN_{local}	46,42	**52,04**	49,06

TABLE 3 – Performances sur la base de validation TAC 2016 en fonction du niveau d'agrégation avec le contexte parfait. Résultats moyennés sur 10 entraînements pour chaque configuration.

du modèle local et que l'agrégation à l'échelle du document soit intrinsèquement meilleure, peut-être en raison d'une possible prévalence des ambiguïtés inter-phrastiques.

Au-delà de la taille du contexte global, sa nature peut être aussi importante. Dans cet article, ce contexte est issu de l'agrégation des prédictions réalisées à une échelle locale. Une approche alternative serait d'utiliser la représentation issue de la couche précédente du modèle ($\mathbf{f_{pooling}}$), représentation plus riche que ses simples prédictions. Nous avons réalisé des expériences préliminaires concernant cette intégration en faisant varier les mêmes paramètres que dans le reste de l'étude (niveau d'agrégation, normalisation, niveau d'intégration). Néanmoins, ces expérimentations ne se sont pas révélées très concluantes, la meilleure configuration (agrégation à l'échelle du document, intégration au niveau du softmax et représentation centrée réduite) obtenant 50,45 et 50,54 en f1-mesure, respectivement en validation et en test.

Concernant le niveau d'intégration du contexte global, les résultats du tableau 2 montrent que les gains de $CNN_{\text{doc-plong\_soft}}$ et de $CNN_{\text{doc-plongement}}$ sont obtenus en privilégiant la précision au détriment du rappel. Au contraire, la configuration la plus favorable, $CNN_{\text{doc-softmax}}$, permet une amélioration de la précision, certes plus faible, mais ne dégradant pas le rappel. Ce modèle étant le plus favorable, nous nous concentrons sur celui-ci dans le reste de cette étude.

4.2 Analyse d'erreur du meilleur modèle

La figure 4 présente un comparatif des performances du modèle local et du modèle $CNN_{\text{doc-softmax}}$. Pour plus de précision, le tableau 4.2 donne en outre les valeurs correspondantes. Une première observation très clairement visible sur la figure 4 est l'existence d'un écart de performance important entre les différentes classes, aussi bien pour le modèle local que global. Puisque notre modèle global s'appuie sur les prédictions du modèle local, on aurait pu craindre que la représentation globale dégrade les performances sur les classes faibles. On constate que ce n'est pas le cas : sur les 5 classes ayant les performances initiales les plus basses, 2 ne sont pas affectées et les performances augmentent pour une classe. Les deux classes restantes, Contact-Contact et Transaction-Transaction, sont les sous-types utilisés en cas d'ambiguïté avec d'autres sous-types des types Contact et Transaction respectivement. On observe dans les deux cas une amélioration d'une autre classe du même type d'événement (respectivement Contact-Correspondance et Transaction-Transfer-money), ce qui indique un probable transfert entre ces sous-types.

Afin d'illustrer qualitativement l'apport de notre méthode, nous présentons deux exemples de prédiction incorrecte par le modèle local, corrigée par le modèle global. La phrase complète est fournie avec le contexte local entouré de crochets et la mention candidate en gras ainsi que les prédictions du modèle local agrégées à l'échelle du document et l'erreur commise par le modèle local.

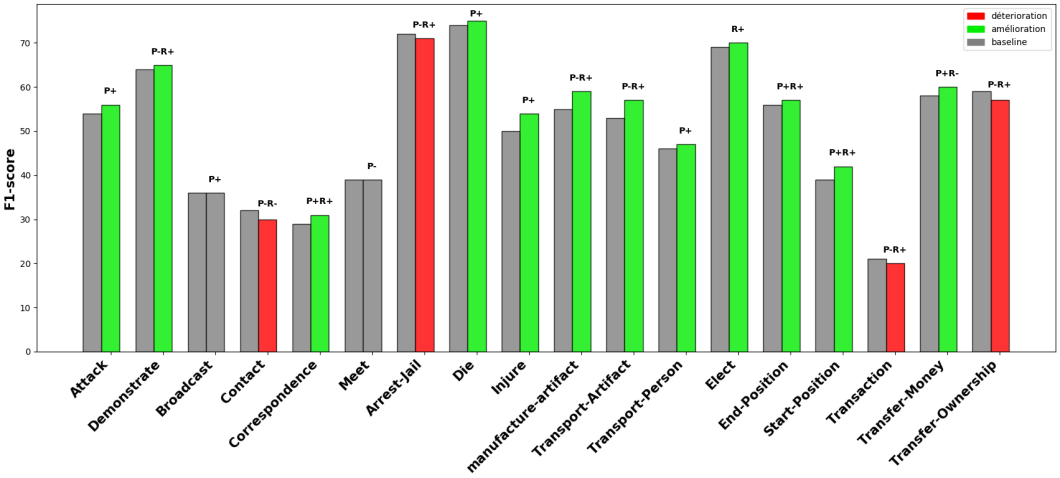

FIGURE 4 – Comparaison des performances par classe entre CNN_{local} et $CNN_{doc\text{-}softmax}$ pour la f1-mesure. Les barres de gauche correspondent au modèle local, celles de droite au modèle global. La barre de droite est verte lorsque l'on observe un gain de f1-mesure pour cette classe, rouge dans le cas contraire et grise en l'absence de variation. Pour les classes présentant une variation, les lettres au dessus indiquent l'origine du changement : **P**récision, **R**appel ou les deux.

— « Do n't get me wrong , I 'm [glad he **won** , I] voted for him . »
Contexte global : (elect : 14, correspondence : 5, contact : 3, transport_person : 3, broadcast : 2)
Faux négatif : elect

— « 100,000 MtGox [Bitcoins were **lost** through theft] (about $ 500 million or 7 % of the outstanding Bitcoins). »
Contexte global : (transfer_ownership : 11, transfer_money : 7, contact_contact : 2, transport_person : 2, manufacture_artifact : 2, die : 1, transaction : 1, arrest_jail : 1)
Faux positif : die

Dans le premier exemple, le contexte local ne permet pas d'identifier clairement que *won* fait référence à un événement de la classe *elect*. Le modèle local ayant détecté de nombreuses autres mentions plus évidentes appartenant à cette classe, le modèle parvient global parvient à identifier le type de la mention. Dans le deuxième exemple, à l'inverse, le modèle local interprète incorrectement *lost* comme faisant référence à un décès. Cependant, les documents faisant référence à des décès contiennent généralement plusieurs mentions de ce type ou d'autres types connexes (*injure*, *attack*). Grâce à cette information notre modèle global ne prédit plus incorrectement cette classe. On notera par ailleurs que dans ce second exemple, même avec un contexte local restreint, il apparaît comme évident que l'on a pas affaire à un événement de type *die*. Ceci met en lumière la forte sensibilité des modèles neuronaux vis-à-vis de la mention candidate, au détriment de son contexte.

Conclusion et perspectives

Dans cet article, nous proposons une nouvelle méthode de représentation du contexte global pour la tâche d'extraction d'événements. Cette méthode est fondée sur l'amorçage. Elle agrège à l'échelle

Type	CNN_{local}			$CNN_{doc\text{-}softmax}$		
	P	R	F	P	R	F
conflict-attack	49	61	54	**52**	61	**56**
conflict-demonstrate	62	66	64	61	**69**	**65**
contact-broadcast	59	26	36	**60**	26	36
contact-contact	25	43	32	24	39	30
contact-correspondence	34	25	29	**38**	**27**	**31**
contact-meet	49	33	39	47	33	39
justice-arrest_jail	63	84	72	62	**85**	71
life-die	70	78	74	**72**	78	**75**
life-injure	47	55	50	**53**	55	**54**
manufacture-artifact	66	47	55	59	**59**	**59**
movement-transport_artifact	75	41	53	74	**46**	**57**
movement-transport_person	43	50	46	**44**	50	**47**
personnel-elect	64	74	69	64	**76**	**70**
personnel-end_position	67	47	56	**68**	**48**	**57**
personnel-start_position	41	38	39	**46**	**39**	**42**
transaction-transaction	32	16	21	24	**18**	20
transaction-transfer_money	54	64	58	**59**	61	**60**
transaction-transfer_ownership	67	53	59	60	**54**	57

TABLE 4 – Comparaison détaillée des performances par classe entre CNN_{local} et $CNN_{doc\text{-}softmax}$. Pour une meilleure visibilité, nous rapportons les mesures sans les décimales. Les performances en **P**récision, **R**appel et **F**-score sont en gras lorsque le modèle global est meilleur que le modèle local, le nom de la classe l'est seulement quand le F-score est meilleur.

du document les prédictions d'un premier modèle local de nature convolutive, obtenant ainsi une représentation du document focalisée sur la tâche finale. Cette représentation est ensuite intégrée à un nouveau CNN. Nous obtenons ainsi des gains significatifs par rapport au modèle local et des performances supérieures, avec un seul modèle, à une association de modèles BiLSTMs.

Notre représentation globale actuelle n'agrège que les prédictions de la couche de sortie du modèle local. Elle souffre donc des imperfections de ce modèle. Pour dépasser cette limite, nous envisageons d'étudier la génération et l'intégration de représentations plus riches tout en restant spécifiques à la tâche. Ces représentations, de nature hiérarchique, pourraient notamment être fondées sur des modèles de classification thématique ou de clustering semi-supervisé.

Remerciements

Ce travail a été partiellement financé par l'Agence Nationale de la Recherche dans le cadre du projet ANR-15-CE23-0018 ASRAEL.

Références

CHEN Y., XU L., LIU K., ZENG D. & ZHAO J. (2015). Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. In *53rd Annual Meeting of the Association for Computational Linguistics and 7th International Joint Conference on Natural Language Proce ssing (ACL-IJCNLP 2015)*, p. 167–176, Beijing, China.

DUAN S., HE R. & ZHAO W. (2017). Exploiting Document Level Information to Improve Event Detection via Recurrent Neural Networks. In *Eighth International Joint Conference on Natural Language Processing (IJCNLP 2017)*, p. 352–361, Taipei, Taiwan.

FENG X., HUANG L., TANG D., JI H., QIN B. & LIU T. (2016). A Language-Independent Neural Network for Event Detection. In *54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, p. 66–71, Berlin, Germany.

GIMPEL K. & SMITH N. (2010). Softmax-Margin CRFs: Training Log-Linear Models with Cost Functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010)*, p. 733–736, Los Angeles, California.

JIANG S., LI Y., QIN T., MENG Q. & DONG B. (2017). SRCB Entity Discovery and Linking (EDL) and Event Nugget Systems for TAC 2017. In *Text Analysis Conference (TAC)*.

KODELJA D., BESANÇON R., FERRET O., LE BORGNE H. & BOROS E. (2017). CEA LIST Participation to the TAC 2017 Event Nugget Track. In *Text Analysis Conference (TAC)*.

LE Q. & MIKOLOV T. (2014). Distributed Representations of Sentences and Documents. In *31st International Conference on International Conference on Machine Learning (ICML 2014)*, p. 1188–1196, Beijing, China.

MAKAROV P. & CLEMATIDE S. (2017). UZH at TAC KBP 2017: Event Nugget Detection via Joint Learning with Softmax-Margin Objective. In *Text Analysis Conference (TAC)*.

MANNING C. D., SURDEANU M., BAUER J., FINKEL J., BETHARD S. J. & MCCLOSKY D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), system demonstrations*, p. 55–60.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. S. & DEAN J. (2013). Distributed Representations of Words and Phrases and Their Compositionality. In *26th International Conference on Neural Information Processing Systems (NIPS 2013)*, p. 3111–3119, Lake Tahoe, Nevada.

NGUYEN T. H., CHO K. & GRISHMAN R. (2016a). Joint Event Extraction via Recurrent Neural Networks. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)*, p. 300–309, San Diego, California.

NGUYEN T. H. & GRISHMAN R. (2015). Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. In *53rd Annual Meeting of the Association for Computational Linguistics and 7th International Joint Conference on Natural Language Proce ssing (ACL-IJCNLP 2015)*, p. 365–371, Beijing, China.

NGUYEN T. H., GRISHMAN R. & MEYERS A. (2016b). New York University 2016 System for KBP Event Nugget: A Deep Learning Approach. In *Text Analysis Conference (TAC)*.

REIMERS N. & GUREVYCH I. (2017). Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, p. 338–348, Copenhagen, Denmark.

SONG Z., BIES A., STRASSEL S., RIESE T., MOTT J., ELLIS J., WRIGHT J., KULICK S., RYANT N. & MA X. (2015). From Light to Rich ERE: Annotation of Entities, Relations, and Events. In *3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, p. 89–98, Denver, Colorado.