Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Représentation dynamique et spécifique du contexte textuel pour l'extraction d'événements

Dorian Kodelja Romaric Besançon Olivier Ferret
CEA, LIST, Laboratoire Analyse Sémantique Texte et Image, Gif-sur-Yvette, F91191 France
dorian.kodelja, romaric.besancon, olivier.ferret@cea.fr

RÉSUMÉ

Dans cet article, focalisé sur l'extraction supervisée de mentions d'événements dans les textes, nous proposons d'étendre un modèle opérant au niveau phrastique et reposant sur une architecture neuronale de convolution de graphe exploitant les dépendances syntaxiques. Nous y intégrons pour ce faire un contexte plus large au travers de la représentation de phrases distantes sélectionnées sur la base de relations de coréférence entre entités. En outre, nous montrons l'intérêt d'une telle intégration au travers d'évaluations menées sur le corpus de référence TAC Event 2015.

ABSTRACT

Dynamic and specific textual context representation for event extraction.

In this paper, which focuses on the supervised detection of event mentions in texts, we propose to extend a neural sentence level model based on graph convolution exploiting syntactic dependencies. To do so, we integrate a larger context through the representation of distant sentences selected on the basis of co-reference relations between entities. We show the interest of such an integration through evaluations carried out on the TAC Event 2015 reference corpus.

MOTS-CLÉS : Extraction d'information événementielle, convolution de graphe, contexte.

KEYWORDS: Event information extraction, graph convolution, context.

1 Introduction

Le travail présenté dans cet article se focalise sur l'extraction d'événements supervisée à partir de textes (Grishman, 2019; Xiang & Wang, 2019; Kodelja *et al.*, 2019b). Cette tâche, incarnée en particulier par les évaluations ACE 2005 (Doddington *et al.*, 2004) et TAC Event (Getman *et al.*, 2018), consiste à identifier dans des textes les mots ou séquences de mots, appelés mentions d'événements, marquant la présence d'un type d'événement défini a priori. Par exemple, le mot « pow-wow » de la phrase suivante :

Putin had invited Tony Blair to the ***pow-wow*** *in Saint Petersburg's Grand Hotel Europe.*

est à extraire pour marquer la présence d'un événement de type *Meet*. Diverses méthodes ont été élaborées au fil du temps pour réaliser cette tâche mais les meilleures performances sont obtenues actuellement par des méthodes présentant deux caractéristiques principales : elles sont fondées sur des architectures neuronales et opèrent principalement à l'échelle phrastique, à l'image de (Nguyen & Grishman, 2018). Néanmoins, se limiter à cette échelle ne permet pas toujours de disposer de

Volume 2 : Traitement Automatique des Langues Naturelles, pages 40–53.
HAL : hal-02784754.

tous les éléments nécessaires à une bonne décision. C'est pourquoi un certain nombre de travaux se sont attachés à exploiter des informations au-delà de la phrase pour extraire de celle-ci des mentions d'événements. Ces travaux peuvent schématiquement se répartir en deux grandes catégories : ceux exploitant des informations au niveau du document pour réaliser une extraction à un niveau local et ceux opérant une extraction plus collective à l'échelle du document.

Les premiers sont représentés historiquement par (Liao & Grishman, 2010) et plus récemment par (Kodelja *et al.*, 2019a), avec une approche par amorçage consolidant au niveau du document les prédictions réalisées à un niveau local afin d'améliorer ces dernières. Sont ainsi prises en compte les dépendances entre types d'événements au niveau du document. (Hong *et al.*, 2011) représente de ce point de vue une vision plus centrée sur les relations entre les types d'entités et les types d'événements, ces relations étant là aussi envisagées à l'échelle du document. Une autre perspective est d'intégrer à un niveau local une représentation globale du document. Duan *et al.* (2017) le font ainsi en s'appuyant sur une méthode de représentation générique, en l'occurrence *Doc2Vec* (Le & Mikolov, 2014). Zhao *et al.* (2018) définissent pour leur part un modèle hiérarchique neuronal de représentation des documents en relation avec la tâche d'extraction d'événements.

Le travail de Chen *et al.* (2018) peut être vu comme une forme d'association entre les deux catégories distinguées ci-dessus : il repose pour une bonne part sur l'exploitation au niveau phrastique des informations présentes à l'échelle du document, exploitation réalisée dans ce cas de façon plus sélective grâce à des mécanismes d'attention, qu'intégratrice, mais l'extraction des événements au niveau phrastique s'effectue de façon collective. Reichart & Barzilay (2012), Liu *et al.* (2016) ainsi que Yang & Mitchell (2016) implémentent des approches collectives opérant quant à elles au niveau du document. La première articule l'utilisation de modèles de type Conditional Random Field (Lafferty *et al.*, 2001) à un niveau local et de contraintes déclaratives à l'échelle du document par le biais d'une procédure de décomposition duale (Rush *et al.*, 2010). La deuxième repose sur le paradigme *Probabilistic Soft Logic* (Kimmig *et al.*, 2012) pour exprimer des contraintes globales sous la forme de formules logiques. Enfin, la dernière s'appuie sur un modèle graphique sous forme de graphe de facteurs (*factor graph*).

Nous proposons dans cet article une nouvelle méthode de prise en compte du contexte textuel pour l'extraction d'événements en étendant un modèle opérant au niveau phrastique fondé sur la convolution de graphe. Cette nouvelle méthode prend en compte de façon sélective le contexte textuel d'une mention d'événement candidate en intégrant la représentation de phrases distantes sur la base des relations de coréférence entretenues avec les entités entourant cette mention. Nous évaluons cette nouvelle méthode sur le corpus de référence TAC Event 2015 et montrons son intérêt en termes de résultats par rapport aux résultats de référence sur ce corpus.

2 Description de l'approche

Comme nous l'avons vu en introduction, l'extraction 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 (Linguistic Data Consortium, 2015) utilisée dans le cadre des campagnes TAC Event (Getman *et al.*, 2018). Ces 38 types sont organisés en 9 grands types pour lesquels nous donnons un exemple entre parenthèses : Business (Merge.Org), Conflict (Attack), Contact (Meet), Justice (Sentence), Life (Divorce), Manufacture (Artifact), Movement (Transport.Person), Personnel (Nominate) et Transaction

(Transfert.Money). Dans les annotations liées à cette taxonomie, les mentions d'événements sont en grande majorité des mots simples. Comme nous pouvons le constater dans le tableau 1, la proportion de mentions d'événements multi-mots pour les corpus d'évaluation que nous avons adoptés se situe aux alentours de 3 %, ce qui est faible. Dans le prolongement de la plupart des modèles neuronaux développés pour l'extraction d'événements, nous choisissons donc d'aborder le problème non pas comme une tâche d'annotation de séquences mais comme une tâche de classification multi-classe de mots, chacun des 38 types d'événements constituant l'une de ces classes, auxquelles s'ajoute une classe NULLE (absence d'événement, classe très majoritaire). 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.*, 2016). Enfin, toujours dans la continuité de la plupart des approches neuronales récentes, nous nous plaçons à l'échelle intra-phrastique. Notre modèle est composé de deux sous-modèles : le premier est un modèle hybride opérant au niveau phrastique tandis que le second, un modèle récurrent, est appliqué au niveau inter-phrastique pour produire une représentation intégrée au premier modèle.

2.1 Modèle phrastique d'extraction d'événements

L'exploitation des informations contenues dans le contexte intra-phrastique joue un rôle capital dans la résolution de la tâche d'extraction d'événements. La compréhension du sens d'un mot en contexte dépend bien évidemment de certains a priori sur son sens en général tels qu'ils sont capturés par des représentations distribuées de mots. Cependant, ce sens contextualisé du mot se sélectionne, voire se construit, par interaction avec le contexte. C'est pourquoi il est nécessaire d'utiliser un modèle exploitant les autres mots de la phrase et la manière dont ils interagissent avec le mot cible. Bien que les modèles exploitant la forme de surface de la phrase (réseaux convolutifs (CNN) ou récurrents (RNN)) permettent d'obtenir des performances satisfaisantes en extraction, la prise en compte de l'arbre de dépendance de la phrase est également bénéfique, comme l'ont notamment montré (Orr *et al.*, 2018) et (Nguyen & Grishman, 2018).

Le modèle de convolution de graphe proposé par (Nguyen & Grishman, 2018) pour l'extraction d'événements ayant obtenu de bonnes performances, nous nous focaliserons par la suite sur cette architecture. Ce modèle est constitué de quatre composantes principales : une couche de plongement associant une représentation vectorielle à chaque mot en entrée spécifiant les différents traits qui lui sont associés ; un BiLSTM (Hochreiter & Schmidhuber, 1997) appliqué à ces représentations d'entrée permettant leur contextualisation dans l'espace de la phrase ; plusieurs couches de convolution de graphe opérant sur ces représentations contextualisées afin de prendre en compte l'environnement des mots par le biais des relations de dépendance syntaxique ; enfin, une couche de pooling permettant d'agréger les représentations ainsi produites pour chaque mot de la phrase avant la couche de classification pour le mot cible. Nous présentons ici plus en détail ces composantes.

2.1.1 Représentation et contextualisation des entrées

Comme nous l'avons vu ci-dessus, la tâche d'extraction d'événements prend la forme d'une classification multiclasse pour chaque mot de la phrase. Pour chaque candidat w_t, nous créons une représentation qui lui est spécifique de la phrase $S = (w_1, w_2, \ldots, w_n)$ dans laquelle il apparaît. Pour ce faire, la représentation x_i de chaque mot w_i est obtenue en concaténant les représentations vectorielles réelles suivantes correspondant aux différents traits associés aux mots en entrée.

— **Plongement de mot :** cette représentation encode les propriétés distributionnelles du mot w_i. Elle est initialisée à partir de représentations produites sur la base d'un grand corpus non annoté.

— **Plongement de position :** pour chaque mot w_i de la phrase, sa distance $i - t$ au mot cible w_t est calculée. Un dictionnaire de vecteurs initialisés aléatoirement associe les différentes distances possibles à des vecteurs.

— **Plongement d'entités :** les entités nommées jouent un rôle important en extraction d'information car elles sont généralement les arguments des relations à extraire. Dans le cas présent, elles sont plus précisément les participants des événements à extraire et sont donc partie prenante, à un niveau plus local, de relations entre les mentions d'événements et les participants de ces événements. Même si toutes les entités nommées d'une phrase ne sont pas nécessairement liées à une mention d'événements présente dans cette phrase, elles constituent néanmoins des indices potentiellement intéressants de sa présence. Du point de vue des représentations, comme pour les plongements de position, chaque type d'entité (dont le type NUL correspondant à l'absence d'entité pour un mot) est associé à un vecteur initialisé aléatoirement. Le type d'entité e_i du mot w_i permet ainsi d'obtenir son plongement d'entité.

La concaténation des représentations x_i de chaque mot w_i d'une phrase forme la séquence X :

$$X = (x_0, x_1, \ldots, x_n) \tag{1}$$

Celle-ci constitue une représentation de la phrase, centrée sur le mot cible w_t grâce aux plongements de position. Cette représentation est ensuite mise en entrée d'un modèle BiLSTM permettant de produire une représentation contextualisée de chaque mot de la phrase. Plus précisément, la partie *forward* de ce BiLSTM intègre la partie de la phrase allant de son début jusqu'au mot considéré tandis que la partie *backward* intègre la partie complémentaire, allant de la fin de la phrase jusqu'au mot considéré. Les réseaux de type LSTM ont vocation à prendre en compte un contexte large, ce qui pourrait sembler a priori suffisant pour couvrir la totalité d'un contexte phrastique. En réalité, l'expérience montre que l'horizon effectif de ce type de réseaux est beaucoup plus limité, ce qui explique d'ailleurs que les performances d'un réseau LSTM en extraction d'événements soient très comparables à celles d'un réseau convolutif, dont l'horizon est supposé plus réduit.

2.1.2 Convolution de graphe pour l'exploitation des dépendances syntaxiques

Afin de dépasser les limites des modèles convolutifs et récurrents, l'idée est de s'appuyer sur les relations de dépendance syntaxique pour élargir l'horizon des modèles de façon sélective. Les participants des événements, qui peuvent constituer des traits particulièrement discriminants pour les identifier, sont en effet fréquemment liés aux mentions d'événements par le biais de relations syntaxiques. Pour mettre en œuvre cette idée, Nguyen & Grishman (2018) ont proposé d'utiliser la notion de convolution de graphe (Kipf & Welling, 2017). Dans ce cadre, le voisinage d'un mot n'est plus constitué de son environnement séquentiel – les mots qui le précèdent et le suivent – mais des mots qui lui sont liés par le biais de relations de dépendance. La représentation d'un mot est alors produite en appliquant une convolution aux représentations des mots constituant ce voisinage.

De façon plus formelle, à partir d'une phrase S de n mots (w_1, w_2,...w_n), est construit le graphe $G = \{V, E\}$ tel que l'ensemble $V = \{w_1, w_2, \ldots, w_n\}$ de ses nœuds correspond aux mots de la phrase considérée et l'ensemble de ses arêtes E, à l'ensemble des relations de dépendance syntaxique

qui les lient. Pour chaque paire de mots (w_i,w_j) pour lesquels w_i et w_j sont respectivement gouverneur et gouverné d'une relation, L(w_i,w_j) indique le type de la dépendance en question.

Afin que la représentation d'un mot par le graphe tienne compte à la fois de la représentation d'entrée du mot, de ses gouvernants et de son gouverneur, l'ensemble des arêtes E est constitué de trois sous-ensembles d'arêtes dirigées et étiquetées.

— **Direct :** pour chaque dépendance syntaxique (w_i,w_j) de type L(w_i,w_j), nous ajoutons une arête de w_i vers w_j étiquetée par le type de la dépendance (p. ex. *nmod*).

— **Inverse :** nous produisons également une arête inverse, de w_j vers w_i, étiquetée par le type de la dépendance et suffixée par l'intitulé *inverse* (p. ex. *nmod_inverse*).

— **Self-loop :** pour chaque nœud du graphe, une arête vers lui-même de type *self-loop* est ajoutée au graphe. Contrairement aux précédents types d'arêtes, celui-ci ne traduit pas une relation syntaxique au sein de la phrase mais permet au modèle de prendre en compte la représentation propre du nœud à la couche précédente lors de la convolution.

Un modèle de convolution de graphe est constitué de K couches de convolutions appliquées à un graphe dont les arêtes peuvent être de différents types. Pour un nœud u de voisinage $N(u)$, sa représentation h_u^{k+1} à la couche $k+1$ est alors :

$$h_u^{k+1} = \sigma\left(\sum_{v \in N(u)} W_{L(u,v)}^k h_v^k + b_{L(u,v)}^k\right) \tag{2}$$

où $W_{L(u,v)}^k$ et $b_{L(u,v)}^k$ sont respectivement la matrice de poids et les biais associés au type de dépendance $L(u,v)$ entre u et v. σ est une fonction d'activation.

Afin de distinguer l'influence de différents voisins, une pondération des nœuds voisins est obtenue ainsi :

$$s_{(u,v)}^k = \sigma\left(h_v^k \overline{W}_{L(u,v)}^k + \bar{b}_{L(u,v)}^k\right) \tag{3}$$

En introduisant la pondération des voisins (3), l'équation de convolution de graphe (2) devient :

$$h_u^{k+1} = \sigma\left(\sum_{v \in N(u)} s_{(u,v)}^k \left(W_{L(u,v)}^k h_v^k + b_{L(u,v)}^k\right)\right) \tag{4}$$

Le nombre de paramètres des matrices $W_{L(u,v)}^k$, $\overline{W}_{L(u,v)}^k$ et des biais $b_{L(u,v)}^k$ et $\bar{b}_{L(u,v)}^k$ est proportionnel au nombre de types de dépendances syntaxiques. Or, les *Universal Dependencies* utilisées sont constituées de 37 dépendances différentes. Le modèle produisant également des dépendances inverses et des arêtes self-loop, il serait nécessaire d'utiliser 75 étiquettes différentes. Compte tenu de la taille relativement petite des jeux de données en extraction d'événements, il est préférable de restreindre le nombre de types de dépendances. Pour ce faire, et comme proposé par (Marcheggiani & Titov, 2017), nous limitons le nombre de relations syntaxiques à trois : lien direct, inverse et self-loop.

Pour la première couche du graphe, la représentation h_u^0 est la représentation contextualisée du mot x_u produite par le BiLSTM. Il est à noter que cette représentation est en pratique complémentaire de l'approche par graphe de convolution : elle permet de prendre en compte d'une façon que l'on sait efficace le contexte local du mot considéré tandis que la vocation du graphe de convolution est surtout de regarder au-delà de ce contexte local.

2.1.3 Pooling

Une fois produite la séquence des représentations vectorielles $h^k_{w_1}, h^k_{w_2}, \dots h^k_{w_n}$ de chaque mot par la dernière (K-ième) couche de convolution de graphe, il est nécessaire d'agréger cette séquence en une représentation p_t du mot cible w_t à fournir en entrée d'une couche linéaire dotée d'un softmax afin de réaliser la classification. Nguyen & Grishman (2018) comparent les méthodes existantes, *pooling cible* (extraction de la représentation du mot cible uniquement), *pooling global* (*max-pooling* sur l'ensemble des mots de la phrase), *multipooling dynamique* (concaténation des poolings globaux des contextes gauche et droit du mot cible) et proposent une nouvelle méthode d'agrégation tenant compte des entités de la phrase : le pooling d'entités.

Cette proposition est motivée par les limites des autres méthodes à tirer spécifiquement profit des représentations vectorielles produites par le graphe des entités. Comme évoqué précédemment, le nombre K de couches de convolution de graphe peut être insuffisant pour que l'information des entités distantes soit propagée jusqu'à la représentation finale h^k_t extraite par la méthode de pooling cible. De plus, la présence d'informations plus spécifiques aux entités, présentes dans leurs représentations propres, est ignorée par cette méthode. Pour ce qui est des deux autres méthodes, leur traitement indifférencié de l'ensemble des mots du contexte peut mener au rejet d'informations pertinentes des entités dans le cas où certaines représentations de mots non informatifs obtiendraient des valeurs plus élevées. Pour éviter ces écueils, la méthode de pooling d'entités consiste à appliquer un *max-pooling* uniquement aux mots cibles et aux entités de la phrase :

$$p_t = maxpool(\{h^K_{w_t}\} \cup \{h^K_{w_i} : 1 \leq i \leq n, e_i \neq \text{NUL}\}) \quad (5)$$

Cette méthode reposant sur une annotation fiable des entités, elle pourrait s'avérer moins efficace dans des cas où l'annotation des entités serait réalisée automatiquement. C'est pourquoi nous proposons un pooling intermédiaire entre le pooling d'entités et le pooling global, ne dépendant pas des entités mais se focalisant également sur les mots les plus porteurs de sens. Cette stratégie, que nous appelons pooling syntaxique, consiste à appliquer un max-pooling au mot cible et à l'ensemble des noms, verbes et adjectifs de la phrase.

2.2 Prise en compte du contexte inter-phrastique

Le modèle de convolution de graphe étant à même d'exploiter un contexte intra-phrastique distant, nous nous intéressons à présent à la prise en compte du contexte inter-phrastique, c'est-à-dire l'exploitation des autres phrases du document pour l'enrichissement de la représentation locale.

2.2.1 Problématique

À notre connaissance, seulement deux autres études se sont portées sur l'intégration d'un contexte distant pour l'extraction d'événements. Ces deux modèles produisent une représentation unique du document, utilisée de manière indifférenciée pour tous les exemples d'apprentissage. Nous faisons au contraire l'hypothèse qu'il est souhaitable de déterminer un contexte spécifique pour chaque exemple afin de produire une représentation du contexte inter-phrastique plus à même de résoudre les ambiguïtés locales spécifiques de la phrase d'exemple.

Dans le cadre de l'extraction d'événements, la présence d'entités communes, en tant qu'arguments potentiels d'événements similaires, nous semble un indice fort du lien contextuel entre deux phrases.

Nous supposons en effet que de telles phrases font référence à des événements proches (tels que différents événements judiciaires partageant un même accusé), successifs (succession d'une blessure puis de la mort), voire contiennent deux mentions d'un même événement.

Nous distinguons ici la notion d'entité, c'est-à-dire une instance unique et spécifique telle qu'une personne, un lieu ou une organisation, de celle de mention d'entité, c'est-à-dire la mention d'une entité au sein d'une phrase. Ainsi, dans les deux phrases suivantes provenant d'un même document du jeu de données ACE 2005 :

— « Putin had invited Tony Blair to the **pow-wow** in *Saint Petersburg*'s Grand Hotel Europe. »

— « But the *Saint Petersburg* **summit** ended without any formal declaration on Iraq. »

le mot « pow-wow » de la première phrase est déclencheur d'un événement *Meet*. Mais ce mot est particulièrement atypique, donc peu susceptible d'avoir été rencontré dans un ensemble d'entraînement. L'identification de la coréférence entre les deux mentions « Saint Petersburg » permet néanmoins de relier cette phrase à la seconde dans laquelle la présence de l'événement est plus évidente. Dans cet esprit, nous proposons de restreindre le contexte d'une phrase cible à l'ensemble des phrases contenant des mentions associées à des entités communes avec la phrase cible. Notre méthode consiste alors à extraire des représentations de ces phrases de contexte puis à les combiner aux représentations locales des mentions d'entités correspondantes de la phrase cible afin de les enrichir. Cette méthode peut donc se diviser en trois étapes que nous présentons dans la suite de cette section : l'identification des phrases de contexte, l'extraction des représentations des entités du contexte puis l'intégration de ces représentations au modèle local.

2.2.2 Sélection des phrases de contexte

À partir d'une phrase $S^j = (w_1^j, w_2^j, \ldots, w_n^j)$, pour chaque mot w_i^j tel que $e_i^j \neq \text{NUL}$, $E(e_i^j)$ désigne l'entité correspondant à la mention e_i^j. Pour deux phrases S^j et S^k, $links(S^j, S^k)$ est l'ensemble des liens d'entités entre les deux phrases.

$$links(S^j, S^k)) = \{(l, m) : E(e_l^j) = E(e_m^k)\} \tag{6}$$

Le contexte d'une phrase S^j est alors donné par $Links(S^j)$. Cette fonction produit l'ensemble des tuples associant une phrase de contexte et les liens qu'elle entretient avec la phrase cible :

$$Links(S^j) = \left\{ \left(S^k, links(S^j, S^k) \right) \ : \ j \neq k \right\} \tag{7}$$

2.2.3 Extraction des représentations du contexte

Nous souhaitons produire des représentations spécifiques pour chaque entité e_i^jde la phrase cible S^j. Nous définissons donc $Ent\text{-}Links(S^j, i)$, qui associe à une mention d'entité de la phrase cible l'ensemble des mentions d'entités en coréférence dans des phrases de contexte :

$$Ent\text{-}Links(S^j, i) = \left\{ (S^k, l) \ : \ S^k \in Links(S^j) \text{ et } E(e_i^j) = E(e_l^k) \right\} \tag{8}$$

Pour chacune de ces paires (S^k, l) du contexte, nous produisons une représentation d'entrée $X^{k,l} = x_1^{k,l}, x_2^{k,l}, \ldots, x_n^{k,l}$ similaire à celle présentée en section 2.1.1, à la différence du vecteur de position.

Ici, pour chaque mot, le vecteur de position exprime cette fois la distance par rapport à la position l de la mention d'entité e_l^k.

Deux méthodes d'extraction sont possibles : le mode *Finale* (éq. 9) consiste à concaténer les représentations finales des deux modèles récurrents tandis que le mode *Mention* (éq. 10) extrait les représentations à l'emplacement de l'entité.

$$\textbf{Finale} : h_{\textbf{contexte}}(w^{k,l}) = [h_{\text{forward}}(x_n^{k,l}); h_{\text{backward}}(x_1^{k,l})] \tag{9}$$

$$\textbf{Mention} : h_{\textbf{contexte}}(w^{k,l}) = [h_{\text{forward}}(x_l^{k,l}); h_{\text{backward}}(x_l^{k,l})] \tag{10}$$

$$h_{\textbf{contexte}}(w^{k,l}) \in \mathbb{R}^{2d_c}$$

où d_c est la dimension de la couche cachée des modèles *forward* et *backward*.

2.2.4 Intégration du contexte

Il est à présent nécessaire d'intégrer les représentations du contexte global de la mention d'entité e_i dans le contexte local. Cette représentation peut être intégrée à deux niveaux : soit sous la forme de plongements supplémentaires lors de la production de la représentation d'entrée, soit sous la forme d'un nœud supplémentaire dans le graphe, voisin de la mention d'entité. Ces deux modes d'intégration sont présentés à la figure 1. Que ce soit pour une intégration au niveau des plongements

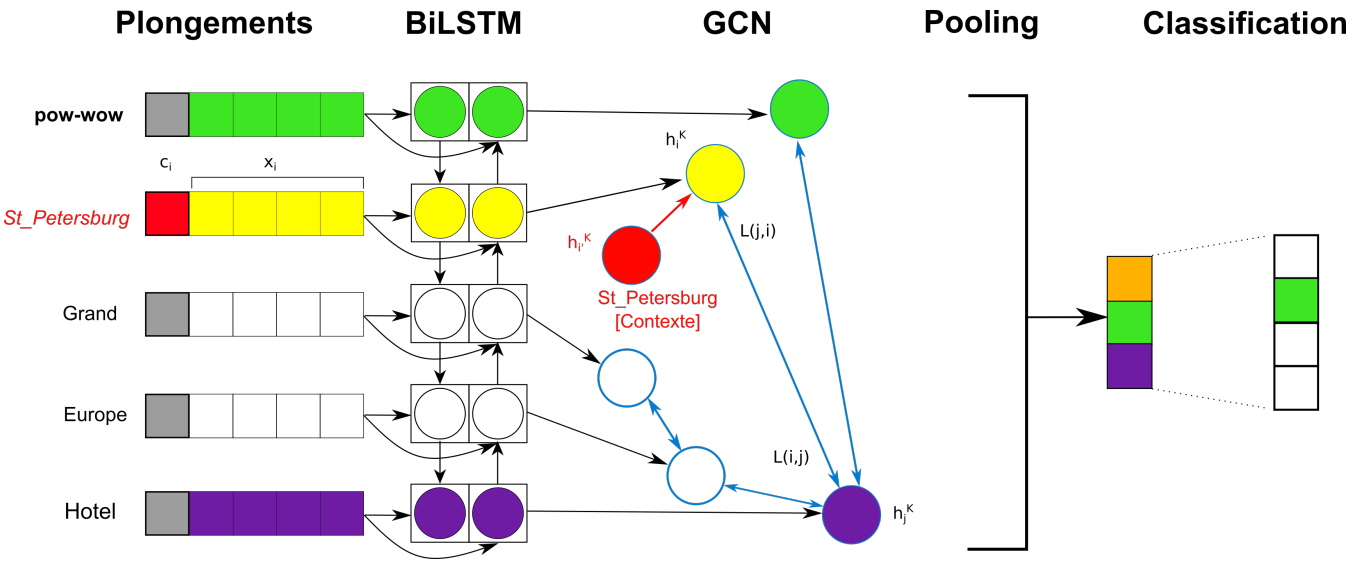

FIGURE 1 – Possibilités d'intégration d'une représentation contextuelle d'entité au sein d'un modèle de convolution de graphe. Cette représentation (en rouge) peut être intégrée en entrée du modèle ou dans le graphe par l'introduction d'un nouveau nœud connecté à la mention locale par un lien de type *contexte*. Le changement de couleur du jaune à l'orange explicite l'influence du contexte sur la représentation finale

ou du graphe, la représentation attendue est un vecteur. Nous agrégeons donc l'ensemble des vecteurs obtenus via une étape de max-pooling permettant d'obtenir le vecteur de contexte [1] :

$$contexte_i = maxpool(\{h_{\textbf{contexte}}(w^{k,l}) \ : \ (S^k, l) \in Ent\text{-}Links(S^j, i)\} \tag{11}$$

1. Par souci de concision et de cohérence avec la notation utilisée en section 2.1.1, nous n'utilisons pas l'index de phrase cible j en exposant de $contexte_i$.

Jeux de données	N^{bre} docs	N^{bre} phrases	N^{bre} mots	N^{bre} évts	N^{bre} évts/doc	% évts multi-mots
train	346	14 568	252 355	7 865	22,7	4,5
dév.	100	6 255	94 064	4 436	44,2	3,7
test	202	4 756	97 956	6 438	31,9	1,8

TABLE 1 – Tailles des jeux de données utilisés : nombre de documents (N^{bre} docs), de phrases (N^{bre} phrases), de mots (N^{bre} mots), de mentions d'événements (N^{bre} évts), nombre de mentions d'événements par document (N^{bre} évts/doc) et pourcentage des mentions d'événements de type multi-mot (% évts multi-mot)

Pour l'intégration au niveau des nœuds, nous modifions donc le graphe $G = \{V, E\}$ en ajoutant un nœud i' dans V associé au mot $w_{i'}$ avec la représentation initiale $h^0_{i'} = contexte_i$ et la représentation finale $h^K_{i'}$. Nous ne créons donc qu'un seul nœud agrégeant l'ensemble des représentations vectorielles distantes. Nous définissons alors un nouveau type d'arête *contexte* pour relier les mentions locales à leur représentation de contexte puis nous introduisons une arête supplémentaire $(w_i, w_{i'})$ de ce type dans E. Pour l'intégration au niveau des plongements, nous concaténons cette représentation aux autres plongements utilisés. Cependant, il est également nécessaire de créer une représentation par défaut pour les mots n'ayant pas de représentation de contexte. Cette représentation $c_{défaut}$ sera modifiée durant l'apprentissage. Le vecteur de contexte (11) est alors généralisé à l'ensemble des mots de la phrase en introduisant :

$$c_i = \begin{cases} contexte_i & \text{si } |Ent\text{-}Links(S^j, i)| > 0 \\ c_{défaut} & \text{sinon} \end{cases} \tag{12}$$

En y introduisant le vecteur (12), nous redéfinissons la séquence d'entrée (1) ainsi :

$$X = ([x_0, c_0], [x_1, c_1], \ldots, [x_n, c_n]) \tag{13}$$

3 Expériences

3.1 Données et prétraitements

Afin de nous comparer aux performances originelles du modèle de graphe présenté par (Nguyen & Grishman, 2018), nous nous évaluons sur le jeu de test TAC 2015. Nous partageons les données d'entraînement de TAC 2015 entre une partie pour l'apprentissage (58 documents) et une partie pour le développement (100 documents) en complétant les données d'apprentissage avec les jeux de données DEFT Rich ERE (R2 V2 et V2). Le tableau 1 présente un certain nombre de statistiques pour ces trois jeux de données : apprentissage (*test*), développement (*dév.*) et test (*test*). Les annotations en entités n'étant pas fournies, nous appliquons un modèle de reconnaissance d'entités nommées pour identifier les mentions. Il est alors nécessaire d'identifier les entités auxquelles ces mentions font référence. Pour ce faire, des outils de désambiguïsation d'entités (*entity linking*) pourraient sembler adaptés. Cependant, ces outils ayant pour objectif de rattacher les mentions d'entités à des entités spécifiques d'une ontologie, ils ne sont pas à même de traiter des mentions telles que « les trois

touristes » ou « l'agresseur », qui sont propres au document considéré. Nous appliquons donc plutôt un outil de résolution de coréférences. Nous redéfinissons alors la notion d'entité comme le groupe de coréférence auquel appartient une mention. Ce processus n'étant pas parfait, certaines mentions sont ignorées. Afin d'élargir la couverture du processus de coréférence, nous fusionnons donc les entités dont les mentions sont identiques.

3.1.1 Génération des exemples

Comme nous l'avons vu lors de la présentation de la convolution de graphe en section 2.1.2, le nombre de couches de convolution K correspond à la distance maximale séparant deux nœuds pouvant mutuellement s'influencer. Afin de faciliter l'accès aux mots supposés porteurs de sens dans la phrase, nous réalisons un filtrage préalable des mots de la phrase en fonction de leur étiquette morphosyntaxique. Nous supprimons ainsi les mots appartenant aux catégories suivantes : ponctuation, symbole, chiffre, déterminant, préposition, conjonction, interjection. Afin de préserver la connexité de l'arbre de dépendances syntaxiques, lorsqu'un mot supprimé est gouverneur d'autres mots, nous remplaçons le gouverneur de ces dépendances par le gouverneur du mot supprimé.

De plus, nous introduisons un masque de prédiction pour ne prédire que pour les noms, les verbes et les adjectifs. Les autres mots sont automatiquement associés à l'étiquette de la classe NULLE. Ce masque permet ainsi de réduire sensiblement le nombre d'exemples négatifs dans le jeu de données en ne perdant que très peu d'exemples positifs. Cette étape présente un double intérêt. D'une part la réduction importante de la taille des jeux de données se traduit par des temps d'apprentissage et de prédiction plus rapides. D'autre part, ce filtrage permet de réduire l'important déséquilibre entre la classe NULLE et les autres classes.

3.1.2 Hyperparamètres

Nous nous appuyons sur la suite d'outils linguistiques Stanford CoreNLP (Manning *et al.*, 2014) pour réaliser l'extraction d'entités nommées, la résolution de coréférences et l'analyse en dépendances utilisée pour produire les graphes. Concernant les entités nommées, nous exploitons l'ensemble étendu de 24 types couvrant des types de base tels que PERSON, LOCATION ou ORGANIZATION mais également des types relevant moins directement de la notion d'entité nommée comme IDEOLOGY ou CAUSE_OF_DEATH. Du point de vue syntaxique, les dépendances considérées sont les dépendances *Basic dependencies*. La matrice de poids des plongements de mots, à 300 dimensions, est initialisée à partir des plongements pré-entraînés GloVe (Pennington *et al.*, 2014). Les plongements de position et de type d'entités sont de taille 50 et les dimensions du BiLSTM du modèle local et des couches de convolution de graphe sont respectivement de 400 et 300. Les plongements de mots, d'entités et de distances sont les mêmes pour les phrases cibles et de contexte. Nous avons repris de (Nguyen & Grishman, 2018) un nombre K de couches de convolution égal à 2, les performances se dégradant à partir de 3, sans doute parce qu'au-delà d'un pas de deux dans le graphe de dépendances, le ratio entre les informations pertinentes observées par rapport à ce que l'on peut considérer comme du bruit vis-à-vis de la tâche devient trop faible. De même, la fonction d'activation σ est aussi la fonction ReLU. Le modèle est entraîné via SGD avec momentum avec des lots de 10 exemples. Les autres paramètres sont déterminés par optimisation bayésienne d'hyperparamètres grâce à hyperopt [2], les

2. https://github.com/hyperopt/hyperopt

	P	R	$F_{max.}$	$F_{moy.}$	F_{σ}
C-GCN	63,39	**57,34**	**60,51**	**60,19**	0,20
Intégration - Nœud	**65,53**	55,40	60,39	59,96	0,38
Pooling - Entité	63,35	57,07	60,49	59,96	0,30
Extraction - Mention	63,11	57,07	60,40	59,86	0,44
Pooling - Cible	62,14	56,92	59,99	59,37	0,36

TABLE 2 – Performances en développement suivant les choix de modélisation (P : précision, R : rappel, $F_{max.}$: F-mesure maximale sur 10 reproductions, $F_{moy.}$: F-mesure moyenne, F_{σ} : écart-type de la F-mesure)

configurations présentées étant sélectionnées à l'aide des performances en développement. Toutes les performances moyennes fournies sont calculées pour 10 reproductions avec les mêmes paramètres.

3.2 Étude des hyperparamètres du modèle

Nous étudions en premier lieu l'influence des différents choix de modélisation présentés :

— **Extraction :** *Finale*/*Mention* (cf. section 2.2.3).

— **Intégration :** *Plongement*/*Nœud* (cf. section 2.2.4).

— **Pooling :** *Cible*/*Syntaxique*/*Entité* (cf. section 2.1.3).

Nous avons réalisé sur le jeu de développement une recherche de valeur optimale pour ces différents paramètres ainsi que pour les paramètres d'optimisation (learning rate, régularisation l2, dropout, momentum). Le meilleur modèle obtenu utilise l'extraction *Finale*, l'intégration *Plongements* et le pooling *Syntaxique*. Cette tendance se confirme également en observant les autres configurations explorées lors de la recherche des meilleures valeurs pour les hyperparamètres. Il n'est cependant pas aisé de résumer directement cet ensemble d'expériences. C'est pourquoi nous présentons dans le tableau 2 les performances du meilleur modèle, C-GCN, et des versions obtenues en modifiant chacun des paramètres présentés. Le pooling cible est très significativement inférieur ($p < 0,001$) au pooling syntaxique utilisé par le modèle C-GCN, ce qui indique que la représentation du nœud à prédire n'est pas suffisante pour en prédire le type. Nous observons également que le pooling des entités obtient des performances légèrement inférieures au pooling syntaxique bien que cette différence soit peu significative ($p = 0,058$). L'exploitation d'un ensemble plus large de mots étant bénéfique au modèle, nous en déduisons que les représentations des entités de la phrase ne suffisent pas à enrichir la représentation du nœud cible. Notre pooling syntaxique est relativement proche du pooling *overall* proposé par Nguyen & Grishman (2018) qui obtient dans l'article d'origine des performances plus basses que le pooling des entités. Puisque nous utilisons un système d'extraction d'entités différent de celui utilisé par Nguyen & Grishman (2018), nous supposons que cette différence de performance est liée à la qualité des entités détectées.

Les moindres performances de l'extraction au niveau des mentions peuvent également s'expliquer par l'imprécision des entités ou simplement par le fait que les représentations finales des phrases de contexte sont plus informatives que les représentations spécifiques des mentions d'entités du contexte. Enfin, concernant l'intégration de la représentation du contexte au modèle, l'intégration au niveau des nœuds ne dégrade pas de manière significative les performances mais produit un profil plus déséquilibré entre précision et rappel.

	P	R	$F_{max.}$	$F_{moy.}$	F_{σ}
GCN_{repro}	78,48	46,96	59,1	58,73	0,82
$C\text{-}GCN_{générique}$	74,50	48,35	59,04	58,64	0,57
C-GCN	75,57	50,42	60,35	**60,47**	0,64
GCN_{nguyen}	70,3	50,6	58,8	-	-
RPI_BLENDER	75,23	47,74	58,41	-	-

TABLE 3 – Performances sur TAC 2015 test

3.3 Comparaison avec l'état de l'art

Nous comparons maintenant notre réimplémentation du modèle de graphe $\mathbf{GCN_{repro}}$ et notre proposition d'extension **C-GCN** à l'implémentation originale $\mathbf{GCN_{nguyen}}$ ainsi qu'au meilleur modèle de la campagne TAC 2015, **RPI_BLENDER** (Hong *et al.*, 2015), fondé sur un classifieur d'entropie maximale utilisant un large ensemble de traits. Afin de confirmer l'intérêt d'un contexte spécifique pour chaque exemple, nous entraînons également $\mathbf{C\text{-}GCN_{générique}}$, exploitant l'ensemble des phrases du document en tant que contexte. Dans ce cas, le vecteur de position n'intervient pas pour les phrases de contexte et la représentation produite sert plongement pour l'ensemble des mots de la phrase cible. Les résultats présentés dans le tableau 3 confirment l'intérêt de notre proposition. En effet, l'introduction de notre représentation de contexte apporte un gain de 1,74 point en F-mesure et permet de dépasser le modèle GCN_{nguyen} détenant jusqu'alors la meilleure performance sur TAC 2015 test, ainsi que le meilleur modèle de la campagne ayant recours à un ensemble large d'attributs définis manuellement. Nous constatons également que l'intégration de la représentation de contexte dans $C\text{-}GCN_{générique}$ ne permet pas d'améliorer les performances par rapport au modèle local, confirmant la pertinence de notre motivation première concernant l'intérêt de fournir un contexte spécifique.

4 Conclusion et perspectives

Nous avons proposé dans cet article une extension d'un modèle de convolution de graphe permettant la prise en compte du contexte inter-phrastique. Cette méthode consiste, à partir d'une phrase cible, à générer une représentation de phrases distantes dans lesquelles apparaissent des mentions d'entités également mentionnées dans la phrase cible. Cette représentation est ensuite utilisée pour enrichir la représentation d'entrée des mentions correspondantes de la phrase cible. L'évaluation de cette méthode sur le jeu de test TAC 2015 permet d'obtenir un gain significatif par rapport au modèle initial, les performances obtenues étant par ailleurs les meilleures sur ce jeu de données.

Notre modèle global étant tributaire de la qualité des étapes d'extraction des mentions d'entités et de leurs liens, une première piste d'extension évidente consiste à étudier leur impact exact sur les résultats en considérant d'autres outils pour l'extraction d'entités nommées et la résolution de coréférences. Par ailleurs, l'utilisation de modèles de désambiguïsation d'entités (*entity linking*) pourrait également étendre les liens entre entités et donc les contextes considérés. De façon complémentaire, substituer au max-pooling utilisé pour agréger les différentes mentions d'entités un mécanisme d'attention permettrait d'apprendre à discriminer et filtrer les phrases de contexte de façon plus fine. Dans le même esprit, un tel mécanisme pourrait aussi être utilisé pour trouver un équilibre plus optimal entre le nombre de couches de convolution et la prise en compte du contexte phrastique par le BiLSTM.

Références

CHEN Y., YANG H., LIU K., ZHAO J. & JIA Y. (2018). Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 1267–1276, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-1158.

DODDINGTON G., MITCHELL A., PRZYBOCKI M., RAMSHAW L., STRASSEL S. & WEISCHEDEL R. (2004). The Automatic Content Extraction (ACE) Program – Tasks, Data, and Evaluation. In *4^{th} Conference on Language Resources and Evaluation (LREC 2004)*, p. 837–840, Lisbon, Portugal : European Language Resources Association (ELRA).

DUAN S., HE R. & ZHAO W. (2017). Exploiting Document Level Information to Improve Event Detection via Recurrent Neural Networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, volume 1, p. 352–361.

GETMAN J., ELLIS J., STRASSEL S., SONG Z. & TRACEY J. (2018). Laying the Groundwork for Knowledge Base Population : Nine Years of Linguistic Resources for TAC KBP. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan : European Languages Resources Association (ELRA).

GRISHMAN R. (2019). Twenty-five years of information extraction. *Natural Language Engineering*, **25**(6), 677—692. DOI : 10.1017/S1351324919000512.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long Short-Term Memory. *Neural Computation*, **9**(9), 1735–1780. DOI : 10.1162/neco.1997.9.8.1735.

HONG Y., LU D., YU D., PAN X., WANG X., CHEN Y., HUANG L. & JI H. (2015). RPI_BLENDER TAC-KBP2015 System Description. In *Proceedings of the 2015 Text Analysis Conference*.

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 : ACL.

KIMMIG A., BACH S., BROECHELER M., HUANG B. & GETOOR L. (2012). A short introduction to probabilistic soft logic. In *NIPS Workshop on Probabilistic Programming : Foundations and Applications*, p. 1–4.

KIPF T. N. & WELLING M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

KODELJA D., BESANÇON R. & FERRET O. (2019a). Exploiting a More Global Context for Event Detection Through Bootstrapping. In *Proceedings of the 41st European Conference on Information Retrieval* , p. 763–770.

KODELJA D., BESANÇON R. & FERRET O. (2019b). Modèles neuronaux pour l'extraction supervisée d'événements : état de l'art. *Traitement Automatique des Langues (TAL), numéro varia*, **60**(1), 13–37.

LAFFERTY J. D., MCCALLUM A. & PEREIRA F. C. N. (2001). Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data. In *Eighteenth International Conference on Machine Learning (ICML'01)*, p. 282–289, Williamstown, MA, USA.

LE Q. & MIKOLOV T. (2014). Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, volume 2 de *ICML'14*, p. 1188–1196, Beijing, China : JMLR.org.

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, Uppsala, Sweden : ACL.

LINGUISTIC DATA CONSORTIUM (2015). *DEFT Rich ERE Annotation Guidelines : Events v.2.6*. Rapport technique.

LIU S., LIU K., HE S. & ZHAO J. (2016). A Probabilistic Soft Logic Based Approach to Exploiting Latent and Global Information in Event Classification. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA : AAAI Press.

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

MARCHEGGIANI D. & TITOV I. (2017). Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 1506–1515.

NGUYEN T. H. & GRISHMAN R. (2018). Graph Convolutional Networks with Argument-Aware Pooling for Event Detection. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA : AAAI Press.

NGUYEN T. H., GRISHMAN R. & MEYERS A. (2016). New York University 2016 System for KBP Event Nugget : A Deep Learning Approach. In *Proceedings of the 2016 Text Analysis Conference*, Gaithersburg, MD, USA : NIST.

ORR J. W., TADEPALLI P. & FERN X. (2018). Event Detection with Neural Networks : A Rigorous Empirical Evaluation. In *Proceedings of the 2018 Conference on Empirical Methods on Natural Language Processing*, Brussels, Belgium : ACL.

PENNINGTON J., SOCHER R. & MANNING C. (2014). Glove : Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, p. 1532–1543, Doha, Qatar : ACL.

REICHART R. & BARZILAY R. (2012). Multi-event extraction guided by global constraints. In *2012 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL HLT 2012)*, p. 70–79, Montréal, Canada.

RUSH A. M., SONTAG D., COLLINS M. & JAAKKOLA T. (2010). On dual decomposition and linear programming relaxations for natural language processing. In *2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, p. 1–11, Cambridge, MA.

XIANG W. & WANG B. (2019). A survey of event extraction from text. *IEEE Access*, **7**, 173111–173137.

YANG B. & MITCHELL T. M. (2016). Joint Extraction of Events and Entities within a Document Context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 289–299, San Diego, California : ACL.

ZHAO Y., JIN X., WANG Y. & CHENG X. (2018). Document Embedding Enhanced Event Detection with Hierarchical and Supervised Attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, p. 414–419, Melbourne, Australia : ACL.