Modèles neuronaux pour l'extraction supervisée d'événements : état de l'art[1]

Dorian Kodelja — Romaric Besançon — Olivier Ferret

CEA, LIST, Laboratoire Analyse Sémantique Texte et Image,
Gif-sur-Yvette, F-91191, France.
{dorian.kodelja,romaric.besancon,olivier.ferret}@cea.fr

RÉSUMÉ. *Cet article de synthèse se situe dans le contexte général de l'extraction d'information et se focalise plus particulièrement sur l'extraction d'événements à partir de textes. Récemment, les approches historiques fondées d'abord sur des règles lexico-syntaxiques puis sur des classifieurs supervisés ont laissé la place à des approches neuronales, à la fois plus intégrées et moins dépendantes de larges ensembles de traits linguistiques extraits* a priori*, ce qui permet de limiter les phénomènes de propagation d'erreurs. Différentes architectures ont été ainsi développées en privilégiant le niveau phrastique, à l'instar des méthodes plus anciennes. Cependant, la complexité de la tâche ne permettant pas de résoudre l'ensemble des ambiguïtés à ce niveau, nous présentons aussi plusieurs approches visant à l'améliorer : approches d'augmentation de données, jointes et globales. Enfin, nous proposons une synthèse des performances des différents choix de modélisation évalués sur le jeu de données ACE 2005.*

ABSTRACT. *This survey takes place in the general context of information extraction and presents more particularly the successive approaches to supervised event extraction from texts. The first rule-based systems and the classical statistical methods use complex and domain-dependent representations that are prone to error propagation. In response to these problems, recent neural network systems using embeddings have linked their success to the absence of the preprocessing steps producing these errors. Among those approaches, different architectures have been proposed to solve the task at the sentence level. However, the task complexity hinders the ability to resolve all ambiguities at this level. Therefore, we identify three ways to enhance the local performance: data augmentation, joint and global inference. Finally, the different design choices presented are compared through an evaluation on the ACE 2005 dataset.*

MOTS-CLÉS : *extraction d'information événementielle, réseaux de neurones.*

KEYWORDS: *event information extraction, neural networks.*

1. Cet article est une version remaniée et étendue de (Kodelja *et al.*, 2017).

1. Introduction

L'extraction d'information est sur un plan général un champ de recherche visant à extraire automatiquement des informations structurées à partir de données textuelles, sources d'informations pas ou peu structurées. Les premiers systèmes d'extraction d'information, développés manuellement pour un besoin précis dans un domaine spécifique, n'étaient en pratique pas réutilisables dans d'autres contextes. Le développement des besoins dans des domaines multiples et impliquant différents types de documents, que ce soient des articles dans le domaine biomédical, des rapports d'inspection dans le domaine industriel ou des dépêches d'agences dans le domaine de la presse, a conduit à la création de systèmes d'extraction d'information de plus en plus modulaires et universels. Cette modularité a naturellement fait apparaître une structuration du processus d'extraction d'information en plusieurs étapes, présentées à la section 2, se caractérisant par l'extraction d'informations de plus en plus complexes. Nous nous focalisons par la suite sur la dernière et la plus complexe de ces étapes d'extraction : l'extraction d'événements. Plus précisément, nous abordons la version supervisée de cette tâche dans laquelle le type des événements à extraire est complètement défini *a priori* et principalement spécifié par le biais d'un ensemble d'exemples annotés dans des textes. Nous nous concentrons sur les développements les plus récents dans ce domaine en lien avec les approches neuronales et présentons les principales architectures développées dans ce cadre, en nous concentrant sur la détection dans les textes des événements et de leurs participants. La plupart de ces travaux s'évaluant dans le même cadre, cette vue d'ensemble s'accompagne d'une comparaison plus quantitative projetant dans une même grille d'analyse les différentes méthodes considérées et permettant de situer les approches neuronales les unes par rapport aux autres, mais également de montrer leur apport vis-à-vis des approches qui les ont précédées. Plus précisément, nous commençons à la section 3 de cet article par donner une vue d'ensemble des différents types de méthodes proposées pour l'extraction d'événements jusqu'à l'émergence des approches neuronales. La section 4 se focalise de façon approfondie sur ces dernières. Ce premier panorama laisse apparaître que la majorité des systèmes d'extraction d'événements se limitent à la prise en compte du contexte local que constitue la phrase. Si ce niveau de contexte est le plus riche du point de vue des analyses linguistiques, il n'est pas toujours suffisant pour résoudre la tâche. L'introduction d'informations supplémentaires peut revêtir différentes formes dans les modèles existants ainsi que le montre la section 5. L'augmentation de données, notamment *via* l'utilisation de ressources externes, est l'une d'elles, étudiée à la section 5.1. La section 5.2 détaille, pour sa part, les modélisations résolvant conjointement différentes tâches d'extraction afin de tirer profit de leurs interdépendances. Outre le fait d'élargir les informations intégrées par chaque tâche, ces approches réduisent ainsi la propagation d'erreurs inhérente aux approches séquentielles. Enfin, les méthodes dépassant le contexte phrastique des mentions événementielles pour l'élargir à un morceau de document, au document entier, voire à d'autres documents, sont présentées à la section 5.3. Les résultats de plusieurs modèles de l'état de l'art appartenant à ces différentes catégories sont présentés puis analysés comparativement à la section 6.

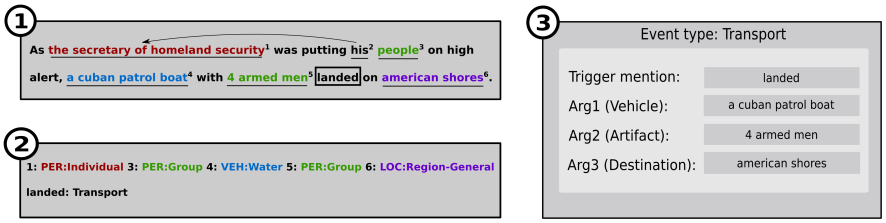

Figure 1. *La reconnaissance d'entités nommées identifie les différentes mentions d'entités (soulignées dans le cadre 1) de la phrase et leur type (cadre 2), mentions auxquelles s'ajoutent les mentions obtenues* via *les liens de coréférence (cf. flèche). La détection d'événements identifie les déclencheurs de la phrase et leur associe un type. Le type du déclencheur indique le type du formulaire du cadre 3. Les arguments de ce formulaire sont ensuite sélectionnés parmi les entités identifiées précédemment.*

2. Définition de l'extraction d'événements

Définir l'extraction d'événements implique logiquement de définir la notion d'événement. Synthétisant l'essentiel des définitions de cette notion en linguistique et en traitement automatique des langues (TAL), Mitamura *et al.* (2015) considèrent ainsi que : « *an event is something that happens at a particular place and time, and it can frequently be described as a change of state* ». Cette définition s'applique assez bien aux événements du jeu de données ACE 2005[1], référence pour l'extraction d'événements présentée à la section 6 et qu'illustrent les quatre types d'événements suivants :

– **Attack** dénote l'action violente d'un *Attacker* à l'aide d'un *Instrument* et induisant des dégâts matériels ou des blessures à une *Target* ;

– **Die** identifie la mort d'une *Victim* causée par un *Agent* à l'aide d'un *Instrument* ;

– **Start-Position** est un événement caractérisé par l'embauche d'une *Person* par un employeur (*Entity*) au poste de *Position* ;

– **End-Position** caractérise à l'inverse la situation d'une *Person* arrêtant d'exercer sa *Position* auprès de l'employeur (*Entity*).

Néanmoins, outre son caractère peu opératoire, cette définition ne couvre que partiellement la notion d'événement telle qu'elle apparaît dans certains travaux. Dans le domaine biomédical, par exemple, la régulation d'une protéine par une autre protéine est vue comme un événement alors que les notions de temps et d'espace ne sont pas prises en compte, le niveau des types d'événements étant le seul pertinent. De ce fait, il apparaît plus générique de considérer par extension un événement comme une forme de relation n-aire caractérisant une configuration d'entités, la notion de changement

1. `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf`

d'état, même si elle est fréquente, pouvant même être absente lorsque l'on considère l'ensemble des métadonnées caractérisant un article de recherche, par exemple (McCallum *et al.*, 2000).

Selon cette optique, installée par les évaluations MUC (Grishman et Sundheim, 1996), l'extraction supervisée d'événements à partir de textes est envisagée comme une tâche de remplissage de formulaire, matérialisant la relation n-aire : le type de formulaire correspond à un type d'événements et impose le remplissage de champs définis *a priori* identifiant les rôles associés à ce type d'événements. La définition de ces rôles s'accompagne de contraintes plus ou moins strictes sur le type des entités susceptibles de les remplir. Ainsi que l'illustre la figure 1, cette extraction d'événements est une tâche complexe, décomposable en plusieurs sous-tâches généralement traitées séquentiellement. Selon le modèle institué par les évaluations ACE (Doddington *et al.*, 2004), ces sous-tâches se répartissent en deux grandes catégories : l'extraction de mentions et l'extraction de liens entre ces mentions.

2.1. *Extraction de mentions*

Reconnaissance d'entités nommées. La première étape d'un système d'extraction d'information consiste à identifier dans le texte l'ensemble des entités pouvant remplir un rôle vis-à-vis d'un événement. Une même entité pouvant apparaître plusieurs fois dans un texte, il s'agit en fait d'extraire des *mentions d'entités*. À la suite des évaluations MUC, trois classes d'entités sont généralement distinguées pour ce qui est du domaine général : les entités désignées par un nom, telles que des personnes, des lieux ou des organisations, les références temporelles telles que les durées ou les dates et les valeurs numériques telles que les prix ou les pourcentages. La définition de contraintes sur les types d'entités par les rôles, comme le fait que la *Victim* d'un événement *Die* ne peut être qu'une *Person*, souligne l'intérêt d'un typage fin des entités, celui-ci restreignant fortement les candidats possibles *a priori*.

Détection d'événements. La majorité des systèmes font l'hypothèse simplificatrice qu'un événement est intégralement défini dans une seule phrase. Ce parti pris est critiquable (Stevenson, 2006), mais motivé par la plus grande richesse des informations exploitables à l'échelle phrastique. La détection d'événements s'assimile alors à la détection de *déclencheurs événementiels* au sein de la phrase, appelés également *mentions d'événements*. La détection d'un événement consiste alors à identifier dans la phrase le ou les mots exprimant le plus clairement la présence d'un événement.

2.2. *Extraction de liens entre mentions*

La tâche d'extraction d'arguments est la principale tâche d'extraction de liens entre mentions que nous considérons ici. Une fois la présence d'un événement d'un type donné identifiée *via* l'extraction d'un déclencheur, il reste à identifier les arguments de cet événement, c'est-à-dire trouver, parmi les mentions d'entités précédemment

extraites, celles se rattachant à l'événement considéré et, le cas échéant, déterminer leur rôle par rapport à lui. Cette tâche est généralement modélisée comme une tâche de prédiction de la relation entre une mention d'entité et le déclencheur.

Sans les développer dans ce qui suit, il faut également souligner l'intérêt de certains liens entre mentions de même type pour l'extraction d'événements : les relations de coréférence lient ainsi plusieurs mentions d'entités faisant référence à la même entité, en particulier lorsque l'une d'elles est de nature anaphorique ; la même logique conduit à identifier les liens entre les déclencheurs faisant référence au même événement, ces relations pouvant aussi être de nature causale ou temporelle.

2.3. *Discussion concernant la modélisation de l'extraction d'événements*

La modélisation de l'extraction d'événements présentée ci-dessus s'est progressivement imposée à la suite des évaluations ACE et a été reprise dans le cadre des évaluations TAC Event (Getman *et al.*, 2018). De ce fait, toutes les approches neuronales développées pour l'extraction supervisée d'événements s'inscrivent, à notre connaissance, dans ce paradigme, dont un certain nombre de points méritent toutefois d'être interrogés. En premier lieu, il faut noter une certaine différence entre le paradigme MUC et le paradigme ACE, même si leur but ultime – le remplissage de formulaire – est identique. ACE met ainsi l'accent sur la détection de déclencheurs événementiels et articule la mise en évidence des participants des événements autour de ces déclencheurs. À l'inverse, MUC met surtout l'accent sur le résultat final à obtenir, c'est-à-dire les formulaires remplis avec les informations extraites des documents, sans ancrage précis de ce qui est extrait de ces documents. Ainsi, la notion de déclencheur événementiel n'est pas présente dans les données MUC et son introduction dans ACE entérine le fait que cette notion, bien que non nécessairement liée à la tâche globale, est considérée comme suffisamment utile à son accomplissement pour la hisser au rang de tâche intermédiaire obligée.

Un des biais de cette conception est de renforcer la focalisation sur la dimension intraphrastique, donc locale, de l'extraction d'événements et de passer sous silence que le peuplement d'un formulaire événementiel ne se limite pas à un ensemble d'extractions locales mais nécessite d'intégrer ces différentes extractions en s'appuyant sur les relations de coréférence évoquées précédemment. Comme nous le verrons à la section 5.3, ce biais ne signifie pas que les approches neuronales ne s'intéressent pas à l'échelle plus globale du document, mais dans ce cas, le document est exploité pour faciliter les extractions au niveau phrastique.

La focalisation sur ce dernier niveau amène également à s'interroger sur la proximité de l'extraction d'événements avec l'étiquetage en rôles sémantiques, en particulier lorsque celui-ci repose sur les cadres sémantiques de FrameNet (Baker *et al.*, 1998). Cette proximité est d'autant plus prégnante que les déclencheurs événementiels sont très majoritairement des verbes et des noms correspondant à des nominalisations, lesquels constituent également la cible de l'étiquetage en rôles sémantiques.

Mais comme le soulignent Abend et Rappoport (2017), les deux tâches sont néanmoins différentes, ce qui explique d'ailleurs une certaine étanchéité entre les travaux les concernant. Cette différence se manifeste, en premier lieu, en termes de granularité : les cadres prédicatifs de l'étiquetage en rôles sémantiques sont généralement d'une granularité plus fine que les événements, qu'il faut plutôt envisager comme des configurations de prédicats en interaction. Dans le même temps, ces cadres sont aussi plus généraux que les événements, en particulier du fait de leur vocation à s'appliquer à tous les prédicats d'un texte. Un événement de type *tremblement de terre*, par exemple, ne correspond pas à un cadre de FrameNet et peut au mieux se ranger sous le cadre très général *Moving_in_place* pour lequel des notions telles que magnitude et épicentre n'existent pas. Enfin, il faut rappeler la dimension discursive, évoquée ci-dessus, de la tâche de remplissage de formulaire événementiel, dimension absente de l'étiquetage en rôles sémantiques. De ce fait, l'appariement entre cadre sémantique et événement ne se fait pas toujours facilement et les rares travaux ayant exploité FrameNet (Liu *et al.*, 2016a ; Chen *et al.*, 2017) se sont en fait contentés d'utiliser les réalisations lexicales de certains cadres pour élargir la liste de leurs déclencheurs.

Enfin, la modélisation de la section précédente présente de façon séparée les différentes tâches de l'extraction d'événements. De fait, ces tâches sont généralement traitées de manière séquentielle, mais des interdépendances existent entre ces différentes étapes et peuvent être exploitées. Les approches jointes concernent généralement la prédiction conjointe de déclencheurs et d'arguments. Les phrases suivantes illustrent l'interdépendance de ces tâches :

1) A cameraman died when an American tank **fired** on the Palestine Hotel.

2) He has **fired** his air defense chief.

Ici, le mot « *fired* » est ambigu et peut indiquer aussi bien un licenciement (*End-Position*) qu'un tir d'arme (*Attack*). Mais, dans le premier exemple, l'entité « tank » correspond de manière évidente au rôle *instrument* d'un événement *Attack*, ce qui permet de déduire qu'il s'agit bien de ce type d'événement. Dans la seconde phrase, puisque « air Defense chief » est un intitulé de poste (*Position*), rôle caractéristique d'un événement du type *End-Position*, la désambiguïsation est évidente.

3. Des origines aux approches neuronales

Le domaine de l'extraction d'information s'est largement développé et structuré au travers de différentes campagnes d'évaluation. Cette particularité s'est manifestée dès l'origine puisque sa naissance est étroitement liée aux évaluations MUC. La tâche de reconnaissance des entités nommées a ainsi été introduite à l'occasion de l'évaluation MUC-6 et la détection d'événements dans le cadre des évaluations ACE. Plus récemment, la piste Event des évaluations TAC KBP a repris pour l'essentiel le cadre fixé par ACE en se focalisant sur la détection d'événements, l'identification de leurs arguments et en y ajoutant la coréférence entre événements. Les jeux de données définis pour cette piste Event n'étant pour le moment pas distribués publiquement, le corpus

ACE 2005 reste la référence utilisée pour évaluer les méthodes les plus récemment développées. Notre comparaison de la section 6 se fera donc sur ce corpus. Si l'extraction d'information s'est principalement définie au travers des campagnes citées, relevant souvent de ce que l'on peut appeler le domaine général, elle a également investi des domaines plus spécialisés. Les campagnes BioCreative[2] dans le domaine de la biologie ou i2b2[3] dans le domaine médical en sont des exemples emblématiques, mais non exclusifs qui ont fourni des jeux de données annotés sur l'identification de relations entre traitements et maladies ou entre protéines et gènes, par exemple.

Tandis que les campagnes d'évaluation ont permis de définir fonctionnellement les contours et le contenu de l'extraction d'information, les méthodes développées pour la mettre en œuvre ont, quant à elles, suivi le mouvement des approches définies plus généralement dans le domaine du TAL. Partant d'approches s'appuyant fortement sur les connaissances et leur représentation, à l'instar du système ATRANS (Lytinen et Gershman, 1986), les travaux ont évolué au début des années 90 vers des approches moins profondes, mais plus extensives, typiquement fondées sur la définition manuelle de patrons lexico-syntaxiques. Le système FASTUS (Hobbs *et al.*, 1997) popularise ainsi lors de MUC-3 l'utilisation d'automates à états finis en cascade et impose dans le même temps une architecture très séquentielle. Par ailleurs, un premier pas vers l'apprentissage est réalisé par des systèmes tels qu'AutoSlog (Riloff, 1993) afin d'acquérir de façon automatique les patrons d'extraction. Dans tous ces travaux, le concepteur définit les représentations et le modèle de réalisation de la tâche cible, l'interprétation de ces deux éléments restant de son ressort.

L'introduction de l'apprentissage statistique constitue une première évolution à cet égard : les représentations, prenant la forme de traits ou descripteurs, demeurent à la charge du concepteur et sont interprétables par lui, mais le modèle de la tâche est construit automatiquement à partir de corpus annotés, typiquement en utilisant un classifieur d'entropie maximale ou des machines à vecteurs de support. Dans le domaine de l'extraction d'information, Zhou *et al.* (2005) ont ainsi introduit pour l'extraction de relations entre entités la plupart des traits utilisés plus largement pour l'extraction d'événements dans des travaux tels que ceux de Li *et al.* (2013). Ces représentations se situent à plusieurs niveaux : lexical (sac de mots et tête de mention des déclencheurs, mots des contextes gauche et droit), syntaxique (chemins dans l'arbre syntaxique entre les deux mentions, *chunking* puis extraction des têtes des groupes nominaux) et sémantique (utilisation des types d'entités ACE et de WordNet (Miller, 1995)). Dans le cas de Li *et al.* (2013), ces traits locaux s'accompagnent de traits plus globaux à l'échelle de la phrase pour intégrer les dépendances entre événements dans une approche jointe.

Dans ces travaux, la représentation des mots, et même plus généralement des traits, est de type parcimonieux ou *one-hot*, prenant la forme d'un vecteur de taille N où N est la taille du vocabulaire et dont seule la dimension correspondant au mot considéré

2. `http://www.biocreative.org/`
3. `https://www.i2b2.org/NLP/`

est active. Cette représentation symbolique pose deux problèmes (Turian *et al.*, 2010) : d'une part, en traitant les mots en tant que symboles discrets et indépendants, les représentations de *courir* et de *coureur* ne sont pas plus similaires que celles de *courir* et *deux*, ce qui ne permet pas aux modèles de capturer la sémantique des mots ; d'autre part, si le vocabulaire de la collection de test est différent de celui de la collection d'entraînement, le modèle ne dispose d'aucune information sur les mots nouveaux.

Une solution à ce problème est, au-delà de l'automatisation de la construction des modèles de réalisation des tâches, d'automatiser également la construction des représentations qu'ils manipulent, à la fois pour améliorer la sensibilité de ces modèles au référent de ces représentations et pour adapter ces dernières aux tâches considérées. Ce type de représentations des mots, appelé plongements lexicaux ou représentations distribuées, a en pratique la forme de vecteurs denses et peut être produit selon différents processus prenant racine dans l'analyse distributionnelle (Harris, 1954) : analyse sémantique latente (Dumais *et al.*, 1988) ou réseaux de neurones implémentant des modèles de langue dans le prolongement de Bengio *et al.* (2003) et popularisés au travers des modèles CBOW et Skip-Gram de Mikolov *et al.* (2013).

4. Architectures neuronales

Le manque d'expressivité des représentations de mots *one-hot*, du point de vue des modèles qui les manipulent, nécessite d'y adjoindre de nombreux traits lexicaux et syntaxiques – catégorie grammaticale, lemme ou appartenance à un lexique spécialisé – pour réaliser une meilleure discrimination. La production de ces traits nécessite la multiplication des étapes de prétraitement et donc la propagation et l'amplification d'erreurs. À l'inverse, les plongements lexicaux semblent capter une partie des informations intéressantes de ces traits tout en s'affranchissant de prétraitements source d'erreurs. En exploitant ces représentations, les modèles neuronaux ont rapidement montré des résultats intéressants en TAL pour des tâches allant de l'étiquetage morphosyntaxique à l'étiquetage en rôles sémantiques (Collobert *et al.*, 2011). Dans ces différents domaines, le succès des approches par apprentissage profond met en avant la capacité d'abstraction et de génération de descripteurs de ces modèles neuronaux.

Classiquement, ces modèles peuvent se décomposer en trois grandes parties : la première, dite *représentation des entrées*, opère sur la forme de l'exemple fourni au réseau et en produit une représentation plus abstraite. Pour ce faire, l'exemple x, constitué ici d'une séquence de m mots, est transformé en une matrice $\mathbf{X} \in \mathbb{R}^{m \times d_{in}}$ en concaténant les vecteurs $\mathbf{x}_w \in \mathbb{R}^{d_{in}}$ associés à chacun des mots x_w, $w \in \{1, m\}$ de l'exemple. L'*extraction de descripteurs* produit ensuite à partir de cette matrice $\mathbf{X}$ un vecteur d'attributs latents $\mathbf{x_{out}} \in \mathbb{R}^{d_{out}}$ servant de base à la classification de l'exemple. Cette représentation finale est alors fournie à la dernière partie du modèle, qui résout la tâche de classification en apprenant à produire un vecteur $\hat{\mathbf{y}} \in \mathbb{R}^{n_c}$ représentant la probabilité d'appartenir à chacune des n_c classes. Cette classification est simplement constituée d'un classifieur linéaire à n_c classes, représenté par une couche dense constituée d'une matrice de poids $\mathbf{W} \in \mathbb{R}^{d_{out} \times n_c}$ et d'un vecteur de

biais $\mathbf{b} \in \mathbb{R}^{n_c}$, qui permet d'obtenir un vecteur de prédictions $\mathbf{o}$, à partir duquel la probabilité d'associer la classe j à l'entrée $\mathbf{x}$ est calculée par une fonction *softmax* :

$$\mathbf{o} = \mathbf{W} \cdot \mathbf{x_{out}} + \mathbf{b} \qquad p(y_j|\mathbf{x}, \theta) = \hat{\mathbf{y}}_j = \frac{\exp^{o_j}}{\sum_{c=1}^{n_c} \exp^{o_c}} \qquad [1]$$

L'apprentissage conjoint de ces trois parties permet l'émergence de représentations spécifiques adaptées à la tâche de classification considérée (Tamaazousti *et al.*, 2019).

Les choix restants pour la modélisation concernant la représentation des entrées et l'extraction de descripteurs portent sur plusieurs aspects : la représentation vectorielle d'un mot, la structure du contexte, l'extraction des descripteurs latents et enfin l'agglomération de ces descripteurs en une représentation vectorielle unique.

4.1. *Représentation des entrées*

Les modèles neuronaux mettent généralement en avant l'intérêt d'une approche bout en bout pour limiter les erreurs, n'utilisant en entrée que les plongements des mots. Néanmoins, pour des tâches complexes comme l'extraction d'événements, d'autres informations y sont généralement adjointes telles que des plongements de position et de types d'entités. Ces plongements complémentaires sont concaténés aux plongements de mots pour former la représentation des entrées.

Types d'entités. L'extraction d'événements venant à la fin de la chaîne d'extraction d'information, la plupart des travaux considèrent la détection des entités du document comme un prérequis déjà satisfait (Nguyen et Grishman, 2015). Cette information peut être ajoutée pour chaque mot par le biais d'un vecteur spécifiant le type d'entités associé au mot en incluant le cas où le mot ne fait pas partie d'une entité. Ce vecteur est construit en initialisant aléatoirement une matrice $\mathbf{W_e} \in \mathbb{R}^{n_e \times d_e}$ avec n_e, le nombre de types d'entités, incluant l'absence de type, et d_e, la taille de leur plongement. Au sein de cette matrice, chacun des types d'entités considérés correspond donc à une ligne dont l'index est utilisé comme identifiant. Cette matrice est ensuite modifiée durant l'apprentissage pour adapter ces plongements en fonction de la tâche.

Position. Dans le cadre des modèles convolutifs utilisant l'agrégation par *max-pooling* (section 4.3), la position des mots dans le contexte est perdue lors de cette dernière étape. Il est donc nécessaire d'introduire directement cette information spatiale dans les vecteurs de mots (Nguyen et Grishman, 2015) pour permettre au système de conserver une information de position relative par rapport au déclencheur, notamment lorsque plusieurs événements sont présents dans la phrase. Pour ce faire, on associe à chaque mot x_w un index p_w correspondant à sa distance avec le déclencheur à l'index t.

$$p_w = w - t, \quad -m < p_w < +m \qquad [2]$$

Tout comme pour les entités, une matrice $\mathbf{W_p} \in \mathbb{R}^{d_p \times (2m-1)}$ initialisée aléatoirement et modifiée pendant l'apprentissage permet d'associer un vecteur à chaque index p_w.

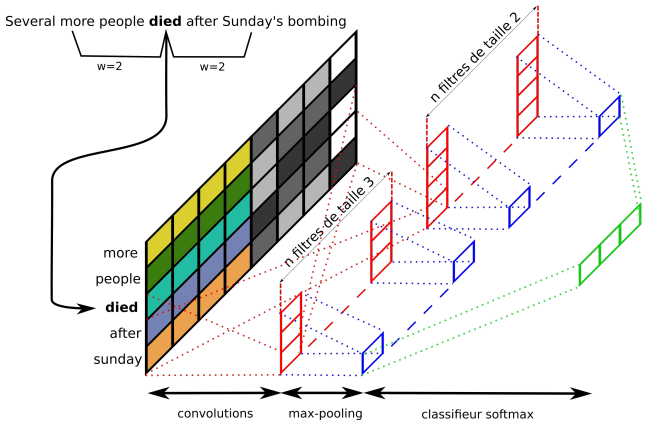

Figure 2. *Schéma de l'architecture d'un CNN*

4.2. *Modélisation de l'extraction de descripteurs*

La force des approches neuronales réside notamment dans la capacité de l'extracteur de descripteurs à identifier au sein des exemples d'entrée un certain nombre d'attributs latents permettant une bonne discrimination du problème considéré. Ces attributs latents sont obtenus par combinaisons non linéaires des vecteurs de mots présents dans le contexte fourni en entrée. Les différentes approches proposées peuvent être organisées selon la nature de la modélisation de ce contexte, qui peut être *séquentielle*, dans l'ordre de la phrase ou *structurée*. Dans les approches séquentielles, la séquence d'entrée est généralement de taille fixe pour des raisons d'implémentation (les séquences plus longues sont dans ce cas tronquées et les séquences plus courtes complétées par un symbole spécial). Ces séquences peuvent être centrées sur le déclencheur candidat (Nguyen et Grishman, 2015) ou alignées sur le début de la phrase (Chen *et al.*, 2015). Pour chaque exemple $\mathbf{x}$, on construit la matrice d'entrée $\mathbf{X} \in \mathbb{R}^{m \times d_{in}}$ par empilement des représentations vectorielles $\mathbf{x}_w$ générées précédemment.

Contexte séquentiel court : architectures convolutives. Les réseaux convolutifs, appelés CNN (*Convolutional Neural Network*) et importés de la vision par ordinateur (Nguyen et Grishman, 2015 ; Chen *et al.*, 2015), exploitent des cooccurrences locales en utilisant une couche de convolution constituée de plusieurs filtres, ainsi que l'illustre la figure 2. Dans le cadre du texte, ces couches convolutives ne comprennent qu'une seule dimension et un filtre de taille k possède un champ récepteur de k mots consécutifs. Plus précisément, dans le cas de la détection d'événements, les mots de chaque phrase sont considérés successivement pour déterminer leur statut éventuel de déclencheur et un exemple $\mathbf{x}$ est représenté par un contexte de taille fixe m centré sur le mot, à l'instar du mot *died* dans la figure 2.

Chaque filtre de convolution de taille k, caractérisé par un vecteur de poids $\mathbf{u_f}$ est appliqué dans l'espace de ce contexte sur une fenêtre glissante de taille k ($k = 2$ dans la figure 2) et génère un vecteur de descripteurs $\mathbf{p_f}$, dont chaque composante est définie par :

$$\mathbf{p_f}[i] = g(\mathbf{x_{i:i+k-1}} \cdot \mathbf{u_f}) \quad [3]$$

où g est une fonction d'activation non linéaire.

Chaque filtre permet ainsi de tirer profit de chaque occurrence de k-grammes au sein du corpus, indépendamment de leurs positions dans le texte. Plusieurs tailles de filtre, typiquement $k \in [2, 5]$, sont généralement utilisées conjointement afin de pouvoir détecter des k-grammes de différentes longueurs. L'utilisation de filtres larges permet également de modéliser des k-grammes à trou (*skip-gram*) de tailles inférieures. Pour étendre cette idée, Nguyen et Grishman (2016) proposent un *Non-consecutive CNN* doté d'une couche de convolution spécifique permettant de calculer l'ensemble des k-grammes à trou de la phrase. Le calcul direct de cette opération à la combinatoire élevée est optimisé à l'aide de la programmation dynamique.

Contexte séquentiel long : architectures récurrentes. Les réseaux convolutifs modélisent les cooccurrences séquentielles locales telles que les k-grammes, mais ne sont pas capables, à l'exception du modèle non consécutif, de gérer des dépendances plus longues et à l'ordonnancement variable. À l'inverse, les modèles récurrents (RNN : *Recurrent Neural Network*) (Nguyen *et al.*, 2016a) sont mieux adaptés à la modélisation des dépendances longues et moins sensibles à la position spécifique des mots. Pour une séquence $\mathbf{x}_{1:m}$, un RNN construit de manière récursive l'état latent (*hidden state*) $\mathbf{s}_w$ d'un mot x_w en fonction de son vecteur d'entrée $\mathbf{x}_w$ et de l'état latent du mot précédent, $\mathbf{s}_{w-1}$:

$$\mathbf{s}_w = \text{RNN}(\mathbf{x}_{1:w}) = \sigma(\mathbf{W_x}\mathbf{x}_w + \mathbf{W_s}\mathbf{s}_{w-1} + \mathbf{b_h}) \quad [4]$$

Cette modélisation est toutefois difficile en raison du problème de l'évanescence du gradient (*vanishing gradient*) : le gradient de l'erreur en provenance de la fin de la phrase s'amenuisant rapidement au cours de la rétropropagation le long de la séquence, il est pratiquement nul pour les premiers mots de la phrase, rendant difficile l'identification des dépendances longues. Pour pallier ce problème, l'architecture *Long Short-Term Memory* (LSTM) (Hochreiter et Schmidhuber, 1997) puis l'architecture *Gated Recurrent Unit* (GRU) (Cho *et al.*, 2014) proposent l'introduction de portes contrôlant la prise en compte de la mémoire et permettant d'éviter ce phénomène.

Si le modèle récurrent et ses variantes LSTM et GRU permettent de modéliser les dépendances longues entre un mot et son contexte passé, le contexte futur de la phrase peut également contenir des informations utiles. C'est pourquoi on utilise généralement une architecture bidirectionnelle (BiRNN), illustrée sur la figure 3, composée de deux RNN indépendants et opposés, nommés *forward* et *backward*. Ces deux composants produisent deux états latents qui sont concaténés pour obtenir la représentation finale du mot, à l'instar du mot *died* de la figure 3.

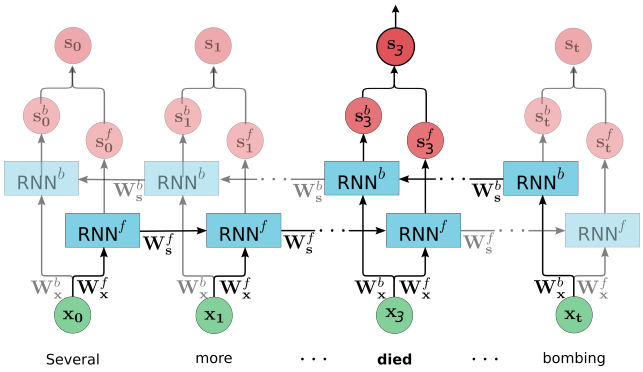

Figure 3. *Schéma de l'architecture d'un RNN bidirectionnel*

Les approches RNN et CNN étant complémentaires, il est également possible de combiner leurs prédictions et d'entraîner conjointement les deux modèles (Feng *et al.*, 2016 ; Nguyen *et al.*, 2016b).

Contexte structuré : architectures de graphes. À l'inverse des approches fondées sur une modélisation séquentielle du texte, les modélisations structurelles s'appuient sur la structure, en général syntaxique, de la phrase. Cette modélisation permet théoriquement de relier plus directement le déclencheur à son contexte et de mieux tenir compte de la variété de ses dépendances. Nguyen et Grishman (2018) introduisent ainsi un modèle de convolution de graphes opérant sur ces seules représentations, sans exploiter le contexte séquentiel. Il est aussi possible d'utiliser cette information en complément de la représentation de surface : Orr *et al.* (2018) modifient un RNN afin d'agréger les états cachés de l'antécédent séquentiel et des antécédents syntaxiques pour conditionner l'état caché du mot courant. De manière similaire, Sha *et al.* (2018) proposent un nouveau LSTM exploitant classiquement la séquence des mots mais doté d'une porte spécifique permettant à l'état caché du mot courant d'être directement conditionné par les mots avec lesquels il est en relation syntaxique.

4.3. *Agrégation des descripteurs : sélection et attention*

Les architectures neuronales présentées précédemment permettent d'extraire des descripteurs, parfois en grand nombre, dans un espace généralement assez large autour du mot à classifier. Il faut alors appliquer une méthode d'agrégation (*pooling*) à ces descripteurs afin de ne conserver que les informations pertinentes pour la classification tout en réduisant la taille de la représentation fournie au classifieur. Nous considérons plus précisément deux approches de ce problème : une approche par sélection des descripteurs et une approche consistant à leur accorder une importance différenciée.

Sélection. Les méthodes d'agrégation par sélection appliquent à la représentation de sortie de l'extracteur de descripteurs une opération ne nécessitant pas l'apprentis-

sage de paramètres supplémentaires. Dans le cas des CNN, les modèles font appel généralement à une fonction de *maxpooling* (Nguyen et Grishman, 2015) conservant la valeur maximale de chaque filtre parmi les valeurs calculées par l'équation 3 :

$$\text{mP}_{[f]} = \max_{1 \leq w \leq m} \mathbf{p_f}_{[w]} \quad \forall f \in [1, n] \qquad [5]$$

L'opération de *pooling* du réseau convolutif ne conservant que l'information prédominante d'une phrase et n'étant pas conditionnée par la position t du déclencheur, il est nécessaire, comme nous l'avons vu précédemment, d'utiliser un plongement de positions pour permettre à la couche de convolution d'apprendre des descripteurs propres au déclencheur d'une part et propres au contexte d'autre part. Chen *et al.* (2015) proposent une variante du CNN utilisant le *dynamic multipooling* (ou *piece-wise CNN* (Zeng *et al.*, 2015)). Dans ce cadre, deux opérations de *pooling* sont appliquées aux deux portions de phrases délimitées par le déclencheur :

$$\text{dmP}_{[f]} = [\max_{1 \leq w \leq t} \mathbf{p_f}_{[w]}; \max_{t \leq w \leq m} \mathbf{p_f}_{[w]}] \quad \forall f \in [1, n] \qquad [6]$$

Le modèle de convolution de graphes de Nguyen et Grishman (2018) utilise l'*entity pooling* appliquant l'opération de *maxpooling* non pas à tous les mots mais uniquement aux représentations du déclencheur et des entités. Dans le cadre des architectures récurrentes, l'état intermédiaire $\mathbf{s_t}$ constitue déjà une représentation du déclencheur x_t conditionnée par son contexte. La méthode dite d'*anchor-pooling*, qui utilise directement cette représentation en entrée du classifieur est introduite dans (Nguyen *et al.*, 2016a).

Attention. La méthode d'*anchor-pooling* repose sur l'hypothèse que cette représentation intermédiaire tient effectivement compte de l'influence des différents mots du contexte sur le déclencheur, ce qui est peu probable pour des dépendances longues. Afin de mieux intégrer le contexte distant, plusieurs mécanismes d'attention ont récemment été proposés. Introduite en traduction automatique dans (Bahdanau *et al.*, 2015), l'attention est utilisée pour l'extraction d'événements (Liu *et al.*, 2018) en attribuant un score de compatibilité r_w entre chaque représentation $\mathbf{s_w}$ du contexte et le déclencheur $\mathbf{s_t}$:

$$r_w = f(\mathbf{s_t}, \mathbf{s_w}) \qquad [7]$$

avec f une fonction non linéaire paramétrée par une matrice de poids apprise durant l'entraînement. Ce score est transformé par un *softmax* pour obtenir l'attention a_w permettant de produire la représentation finale $\mathbf{x_{out}}$ du déclencheur :

$$\mathbf{x_{out}} = \sum_{w=0}^{m} a_w \mathbf{s_w} \qquad a_w = \frac{\exp^{r_w}}{\sum_{j=0}^{m} \exp^{r_j}} \qquad [8]$$

Les mécanismes d'attention imposent l'introduction de poids supplémentaires, ce qui peut s'avérer rédhibitoire ou limitant en l'absence de données suffisantes. Afin d'améliorer l'apprentissage du modèle, il est possible d'utiliser des connaissances externes pour entraîner l'attention de manière supervisée. Liu *et al.* (2017) proposent

ainsi de définir manuellement un vecteur d'attention de référence avec ${r_w}^*$ valant 1 si le mot est un argument du déclencheur et 0 dans le cas contraire. Il est alors possible de définir une nouvelle fonction de coût pénalisant la différence entre le vecteur d'attention produit par le modèle et le modèle de référence.

5. Enrichir le contexte local

Les différentes modélisations présentées à la section précédente peuvent être vues de manière chronologique comme des optimisations successives de la prise en compte du contexte phrastique, les différentes architectures d'extracteurs et d'agrégateurs de descripteurs visant à exploiter des dépendances plus longues ou au contraire à modéliser la tâche pour réduire la distance aux informations contextuelles pertinentes. Cependant, ces modélisations s'avèrent toujours insuffisantes pour résoudre pleinement la tâche de détection supervisée d'événements. Plusieurs facteurs peuvent expliquer ces limites. Tout d'abord, la définition de la tâche est suffisamment ambiguë pour que le maximum théorique ne soit intrinsèquement pas atteignable. Les résultats obtenus par des annotateurs humains, évalués par Hong *et al.* (2011) et présentés dans le tableau 2, vont d'ailleurs dans ce sens. D'autre part, si les approches neuronales actuelles permettent de s'affranchir de traits linguistiques définis manuellement, elles nécessitent, du fait d'un nombre de paramètres à apprendre très important, des volumes de données bien plus importants que ceux disponibles actuellement. Enfin, la plupart des approches actuelles opèrent au niveau phrastique et ne peuvent pas exploiter de contexte plus global en cas d'ambiguïtés locales. C'est pour répondre à ces limites que plusieurs extensions des modèles locaux ont été proposées récemment afin d'enrichir les informations prises en compte par ces derniers. Nous distinguerons ici les approches d'augmentation de données, visant à engendrer de nouveaux exemples d'apprentissage ou à les enrichir, les approches jointes, visant à exploiter la complémentarité entre différentes tâches d'extraction, et enfin les approches globales, visant à exploiter des informations au-delà du contexte phrastique des déclencheurs.

5.1. *Augmentation de données*

Volume. L'augmentation de données en volume consiste à produire automatiquement de nouveaux exemples d'apprentissage. Chen *et al.* (2017) utilisent Freebase (Bollacker *et al.*, 2008) et FrameNet pour extraire ces nouveaux exemples à partir de Wikipédia en amont de l'apprentissage. Leur méthode permet d'obtenir dix fois plus de données que le corpus d'origine et fournit un gain particulièrement significatif. Liu *et al.* (2018) exploitent, pour leur part, les progrès des systèmes de traduction automatique pour engendrer de nouvelles données tout en contournant certaines ambiguïtés monolingues. L'approche traduit automatiquement un corpus anglais en chinois puis utilise un outil d'alignement pour détecter la projection des déclencheurs dans les phrases chinoises. Un extracteur de descripteurs est appliqué pour chaque langue puis leurs représentations sont combinées à l'aide d'un mécanisme d'attention. Hong *et al.*

(2018) proposent d'employer un réseau antagoniste génératif (*generative adversarial network*), c'est-à-dire un modèle générant de manière non supervisée de nouveaux exemples d'apprentissage conçus pour piéger un système discriminant. En entraînant conjointement les deux modèles, le système discriminant est poussé à identifier des caractéristiques plus robustes pour la prédiction.

Richesse. Indépendamment de l'augmentation du nombre d'échantillons d'apprentissage, il est possible d'enrichir chaque échantillon à l'aide d'attributs supplémentaires. Hong *et al.* (2011) démontrent ainsi l'intérêt de produire par regroupement des sous-types plus fins pour les entités. La méthode s'appuie sur l'utilisation de requêtes à des moteurs de recherche, ce qui rend les calculs longs et le passage à l'échelle limité par les restrictions des API de recherche. Liu *et al.* (2016b) proposent un algorithme similaire sans utiliser de telles requêtes. Pour chaque type d'entités, WordNet permet d'associer à chaque mention d'entité des traits supplémentaires (hyperonymes, synonymes) utilisés ensuite par un algorithme de regroupement. Cette procédure permet d'obtenir des sous-catégories plus informatives telles que *président* pour les *personnes* ou *ville* pour les *lieux*. Zhang *et al.* (2018) exploitent enfin les relations entre entités en les encodant à l'aide de plongements d'entités fournis en entrée du modèle.

5.2. *Approches jointes*

La plupart des modèles actuels réalisent de manière jointe la détection et la classification des déclencheurs et il est acquis que ces deux tâches doivent être réalisées conjointement (Chen et Ng, 2012 ; Kodelja *et al.*, 2019). Historiquement, les approches jointes font plus généralement référence à l'extraction conjointe des déclencheurs et des arguments associés. Dans ce cadre, le modèle *sentRules* de Grishman *et al.* (2005) extrait automatiquement différents patrons liant déclencheur et arguments sur le jeu d'apprentissage pour les appliquer en test. Li *et al.* (2013) définissent une méthode évaluant à l'aide d'un perceptron structuré les différentes combinaisons d'assignation de déclencheurs et d'arguments en utilisant un algorithme de recherche par faisceau pour réduire la complexité de l'inférence. Il permet de considérer l'intégralité des interactions entre les différents déclencheurs et arguments. Plus récemment, Nguyen *et al.* (2016a) entraînent de manière jointe deux classifieurs à l'aide de la log-vraisemblance du modèle joint pour une extraction des déclencheurs et des arguments reposant sur un modèle BIGRU commun. Enfin, Sha *et al.* (2018) proposent d'entraîner deux classifieurs conjointement à l'aide de la version structurée de la *hinge loss*. Bien que ces approches jointes puissent bénéficier aux deux tâches, on constate généralement que l'intérêt pour la détection de déclencheurs est négligeable, contrairement aux gains observés pour l'extraction des arguments. Ceci s'explique dans le cas des deux approches précédentes à base de classifieurs par leur fonctionnement interne séquentiel. Si les paramètres des modèles sont optimisés de manière jointe durant l'apprentissage et peuvent donc tirer profit des interdépendances, la prédiction est, quant à elle, séquentielle : les arguments sont évalués uniquement en fonction de la détection et de la classification d'un déclencheur et ne permettent pas, à l'inverse, de modifier

les prédictions des déclencheurs. L'extraction jointe des entités et des événements est également possible (Yang et Mitchell, 2016), bien que la plupart des systèmes actuels considèrent l'extraction des entités comme résolue *a priori* et utilisent des annotations de référence ou un outil externe d'annotation en entités nommées.

5.3. *Prise en compte du contexte global*

Indépendamment de la taille du contexte global considéré, on distingue deux approches pour prendre en compte ce contexte.

Inférence globale. La première approche consiste généralement à filtrer et propager des prédictions réalisées par un modèle local afin de maximiser *a posteriori* la cohérence de ces prédications à une échelle plus globale (Yangarber et Jokipii, 2005 ; Jean-Louis *et al.*, 2011). Ji et Grishman (2008) font ainsi l'hypothèse de la cohérence globale entre mentions et types d'événements (ou de rôles) pour améliorer les prédictions du modèle *sentRules* : dans un même contexte (document ou cluster de documents), un mot sera toujours déclencheur d'un même événement ou y tiendra toujours le même rôle. Pour chaque type d'événements, un ensemble de règles est appliqué pour obtenir une cohérence globale au niveau du document puis pour un cluster de documents obtenu de manière non supervisée. Pour ce faire, les mentions marginales sont filtrées et les mentions fréquentes propagées au sein du contexte considéré. Liao et Grishman (2010) développent cette idée en exploitant la cohérence interévénement pour les déclencheurs et les arguments : la détection d'un événement *Start-Position* augmente la probabilité d'observer un événement *End-Position* tandis qu'une entité tenant un rôle de *Victim* d'un événement *Die* a une probabilité importante d'être *Target* d'un événement *Attack*. Cette cohérence est exploitée en filtrant les prédictions du modèle *sentRules* : seules les prédictions dont la confiance est supérieure à un seuil sont conservées. Les statistiques ainsi générées à l'échelle des documents sont utilisées par un second modèle statistique. L'architecture jointe de (Li *et al.*, 2013) peut également être considérée comme une approche d'optimisation globale, mais au niveau de la phrase : contrairement aux autres modèles dits locaux, l'optimisation est en effet réalisée au niveau phrastique et non pas individuellement pour chaque déclencheur. Enfin, Liu *et al.* (2016b) apprennent une régression logistique sur un ensemble d'attributs locaux et latents pour estimer une première probabilité de classification des déclencheurs avant d'employer un modèle PSL (*probabilistic soft logic*) considérant les cooccurrences entre événements à plusieurs niveaux pour optimiser la prédiction à l'échelle du document. Le modèle prend également en compte les dépendances entre événement et thème en utilisant l'allocation de Dirichlet latente (Blei *et al.*, 2003).

Plongement de contextes. À l'inverse des modèles présentés jusqu'à présent, optimisant la cohérence globale des prédictions, ces approches neuronales considèrent généralement l'information globale du document comme un attribut permettant d'enrichir le modèle local. Duan *et al.* (2017) proposent ainsi d'utiliser un modèle général de plongement de documents, *doc2vec* (Le et Mikolov, 2014), pour obtenir de manière non supervisée le plongement des documents traités, fourni ensuite en entrée d'un mo-

dèle local. Selon la même motivation, Kodelja *et al.* (2019) réalisent un apprentissage en deux passes : les prédictions d'un premier modèle entraîné au niveau local sont agrégées et constituent une représentation du document spécifique à la tâche fournie à un nouveau modèle. Cette seconde passe permet de maximiser la cohérence globale du document. Enfin, Zhao *et al.* (2018) utilisent un modèle hiérarchique de documents permettant de produire un plongement des documents qui, contrairement à Duan *et al.* (2017), est conditionné spécifiquement par la tâche d'extraction d'événements.

6. Comparaison

Cadre. Afin d'étudier l'apport des différents choix de modélisation, le tableau 2 présente les performances des différents modèles introduits précédemment sur le jeu de données ACE 2005. Ce corpus se compose de 599 documents provenant de différentes sources : des dépêches d'agence de presse (106), des bulletins (226) et débats télévisés (60), des blogs (119) et groupes de discussion en ligne (49) et enfin, des transcriptions d'échanges téléphoniques (39). Cette pluralité de sources a conduit à imposer ACE 2005 comme un cadre de référence pour l'extraction d'événements, permettant aussi de tester certaines formes d'adaptation au domaine pour cette tâche (Nguyen et Grishman, 2015). Suite à (Ji et Grishman, 2008), un découpage s'est imposé, avec 529 documents (14 849 phrases et 4 420 déclencheurs) de différentes sources pour l'apprentissage, 40 dépêches (672 phrases et 424 déclencheurs) pour le test et 30 documents de différentes sources (836 phrases et 505 déclencheurs) pour la validation. La tâche de détection d'événements couvre 6 types d'événements et 33 sous-types, avec une difficulté notable : 1 543 occurrences pour le sous-type le plus fréquent, *Attack*, mais 2 occurrences pour la moins fréquente, *Pardon*, d'où un jeu de données assez déséquilibré. D'un point de vue plus linguistique, on peut noter que les déclencheurs se répartissent pour l'essentiel de façon équilibrée entre deux grandes catégories morphosyntaxiques : 46 % de noms pour 45 % de verbes. Les déclencheurs sont, en outre, très majoritairement des termes simples, avec seulement 4 % de multitermes. De ce fait, la détection d'événements est presque toujours abordée comme une tâche de classification de mots, en laissant de côté le problème des multitermes. Enfin, ACE 2005 distingue 34 rôles pour les arguments. Les types d'événements peuvent avoir de 2 à 7 rôles et le plus souvent autour de 5. Ces rôles sont occupés par des entités au sens de la section 2, qui sont soit générales, comme des personnes ou des organisations, soit plus spécifiques, comme des armes ou des véhicules.

Résultats. Les spécificités des modèles du tableau 2 sont synthétisées dans le tableau 1 avec les notations suivantes pour désigner leurs différentes caractéristiques :

– **attributs** *word* : mot, *lex* : lexicaux, *syn* : syntaxiques, *ets* : type d'entités, *ets+* : entités fines, *NER* : entités nommées extraites, *brown* : clusters de Brown, *sg-(nyt/gn/g/t8)* : plongements Skip-Gram entraînés sur le corpus (NYT/Google News/Gigaword/text8), *ccbow/elmo* : autres plongements, *dist* : plongements de distance, *deps* : dépendances, *re* : plongements de relations ;

– **contexte** *seq/graphe* : modélisation séquentielle ou structurée de la phrase ;

– **agreg.** *(Max/Dyn/ety)P* : *maxpooling* ou *dynamic multipooling* ou *entity pooling*, *(s/u)Att* : attention (supervisée ou non supervisée) ;

– **augment.** *sim* : clusters de documents similaires, *wiki* : extraction Wikipédia, *ets+* : entités fines, *trad* : apprentissage multilingue, *re* : plongements de relations ;

– **joint :** *evt(,ety)* : prédiction jointe des déclencheurs (entités) et arguments ;

– **global :** *inf-(evt/sent/doc/docs)* : inférence globale à l'échelle de (l'événement, de la phrase, du document ou du cluster de documents), *d2v* : doc2vec, *HDE* : plongement hiérarchique de documents.

Analyse. Comme on peut le constater au niveau du tableau 1, les paramètres sont assez différents d'un modèle à un autre, ce qui rend difficile toute conclusion définitive. L'analyse que nous ferons ici permettra donc seulement de dégager quelques tendances globales. Tout d'abord, d'un point de vue chronologique, on observe l'arrivée puis la prédominance des approches neuronales à partir de 2015. Cette arrivée coïncide avec l'expansion plus générale de ces modèles dans le domaine du TAL. Si les premiers travaux les concernant mettaient en avant la capacité à s'appuyer uniquement sur les données brutes, sans recourir à des prétraitements linguistiques (Nguyen et Grishman, 2015 ; Chen *et al.*, 2015 ; Feng *et al.*, 2016), on voit progressivement réapparaître l'emploi de ces prétraitements, notamment avec l'utilisation de dépendances syntaxiques pour enrichir les représentations (Nguyen *et al.*, 2016a) ou les structurer dans les approches à base de graphes. En outre, le recours à des ressources externes comme FrameNet ou Freebase pour l'augmentation de données (Chen *et al.*, 2017) vient également mitiger cette volonté initiale de limiter les prétraitements. Sans surprise, l'ajout des entités (CNN_{ets}) permet de mieux résoudre la tâche d'extraction de déclencheurs, avec un apport de 1,4 point pour un CNN. Dans le cas de Yang et Mitchell (2016), il n'est pas évident de déterminer si le gain est dû à l'optimisation jointe des entités et des événements ou au fait que cette optimisation se fasse de manière globale. Les premières approches récurrentes, utilisant simplement l'*anchor-pooling*, motivaient ce choix d'architecture par la capacité du modèle à prendre en compte un contexte plus large et exploiter des dépendances plus longues. Or, les architectures convolutives et récurrentes classiques semblent être équivalentes, DMCNN, CNN, CNN_{ets}, jointBIGRU, $BILSTM_b$ et BiGRU obtenant tous des scores entre 69 et 69,3 en f1-mesure, avec toutefois des profils en termes de précision et de rappel variant grandement. Il semble donc que cette prémisse soit contestable et que les modèles se focalisent en pratique sur un contexte relativement proche (Kodelja *et al.*, 2019). Les architectures convolutives et récurrentes semblent cependant apprendre des représentations complémentaires comme en témoigne le gain obtenu par le modèle Hybrid. En permettant au modèle convolutif local d'exploiter un contexte plus large, la convolution non consécutive de Nguyen et Grishman (2016) permet d'obtenir un gain de 2,3 points par rapport à CNN_{ets} tandis que la majorité des modèles récurrents récents (GMLATT, $BILSTM_{re}$, DEEB, JMEE) utilisent un mécanisme d'attention pour mieux capter l'interaction entre le déclencheur et le reste de la phrase. Les récentes architectures à base de graphes semblent également prometteuses pour une meilleure prise en compte du contexte au niveau phrastique.

Référence	Identifiant	Local				Extension		
		attributs	contexte	extracteur	agreg.	augmt.	joint	glob.
(Grishman *et al.*, 2005)	sentRules	word,lex,synt,ets,NER,	-	-	-	-	evt	inf-evt
(Ji et Grishman, 2008)	crossSents	sentRules	-	-	-	-	-	inf-doc
	crossDocs	sentRules	-	-	-	sim	-	inf-docs
(Liao et Grishman, 2010)	crossEvents	sentRules	-	-	-	-	-	inf-doc
(Hong *et al.*, 2011)	crossEntity	word,ets+	-	-	-	ets+	-	inf-doc
(Li *et al.*, 2013)	jointStruct	word,lex,synt,ets,brown	-	-	-	-	evt	inf-sent
(Chen *et al.*, 2015)	DMCNN	sg-nyt,dist	seq	CNN	DynP	-	-	-
(Nguyen et Grishman, 2015)	CNN	sg-gn,dist	seq	CNN	MaxP	-	-	-
	CNN_{ets}	sg-gn,dist,ets	seq	CNN		-	-	-
(Yang et Mitchell, 2016)	withinSent	word,sg-gn,lex,synt,NE	graphe	factor graph		-	evt	-
	JointEvtEty	word,sg-gn,lex,synt,NE	graphe	factor graph + CRF		-	evt,ety	
(Nguyen *et al.*, 2016a)	jointBIGRU	ccbow-gw,ets,deps	seq	GRU	anchor	-	evt	-
(Nguyen et Grishman, 2016)	NC-CNN	sg-gn,dist,ets	seq	NCCNN	MaxP	-	-	-
(Feng *et al.*, 2016)	$BILSTM_a$	sg-nyt	seq	LSTM	anchor	-	-	-
	Hybrid	sg-nyt,dist	seq	LSTM/CNN	anchor	-	-	-
(Liu *et al.*, 2016b)	PSL global	words,ets+	-	-	-	ets+	evt	inf-doc
(Chen *et al.*, 2017)	$DMCNN_{DS}$	sg-nyt,dist	seq	CNN	DynP	wiki	-	-
(Liu *et al.*, 2017)	ATT	sg-nyt,ets	-	-	sAtt	-		-
	ATT_{DS}	sg-nyt,ets	-	-	sAtt	wiki		-
(Duan *et al.*, 2017)	$BILSTM_b$	sg-nyt	seq	LSTM	anchor	-	-	-
	$BILSTM_{d2v}$	sg-nyt	seq	LSTM	anchor	-	-	d2v
(Liu *et al.*, 2018)	GMLATT	sg-nyt,ets,dist	seq	GRU	uAtt	trad	-	-
(Zhang *et al.*, 2018)	$BILSTM_{Re}$	sg-nyt,ets,re	seq	LSTM	uAtt	re	-	-
(Nguyen et Grishman, 2018)	graphCNN	sg-gn,dist,ets,deps	graphe	graphCNN	etyP	-	-	-
(Zhao *et al.*, 2018)	BIGRU	sg-gn,ets	seq	GRU	anchor	-	-	-
	DEEB	sg-gn,ets	seq	GRU	anchor	-	-	HDE
(Orr *et al.*, 2018)	DAG-GRU	elmo,deps	graphe	DAG-GRU	uAtt	-	-	-
(Sha *et al.*, 2018)	JMEE	sg-t8,	graphe	DBLSTM	anchor	-	evt	-
(Hong *et al.*, 2018)	SELF	sg-nyt,ets	seq	LSTM+GAN	anchor	-	-	-

Tableau 1. *Modèles comparés sur le jeu de test ACE 2005*

Identifiant	Identification déclencheur			Classification déclencheur			Identification argument			Classification argument		
	p	r	f	p	r	f	p	r	f	p	r	f
sentRules	-	-	-	67,6	53,5	59,7	47,8	38,3	42,5	41,2	32,9	36,6
crossSents	-	-	-	64,3	59,4	61,8	54,6	38,5	45,1	49,2	34,7	40,7
crossDocs	-	-	-	60,2	76,4	67,3	55,7	39,5	46,2	51,3	36,4	42,6
crossEvents	-	-	-	68,71	68,87	68,79	50,85	49,72	50,28	45,06	44,05	44,55
crossEntity	n/a	n/a	n/a	72,9	64,3	68,3	53,4	52,9	53,1	51,6	45,5	48,3
HUMAIN	-	-	-	74,3	76,2	75,24	68,5	75,8	71,97	61,3	68,8	64,86
jointStruct	76,9	65,0	70,4	73,7	62,3	67,5	69,8	47,9	56,8	64,7	44,4	52,7
DMCNN	80,4	67,7	73,5	75,6	63,6	69,1	68,8	51,9	59,1	62,2	46,9	53,5
CNN	-	-	-	71,9	63,8	67,6	-	-	-	-	-	-
CNN_{ets}	-	-	-	71,8	66,4	69,0	-	-	-	-	-	-
withinSent	76,9	63,8	69,7	74,7	62,0	67,7	72,4	37,2	49,2	69,9	35,9	47,4
JointEvtEty	77,6	65,4	71,0	75,1	63,3	68,7	73,7	38,5	50,6	70,6	36,9	48,4
jointBIGRU	68,5	75,7	71,9	66,0	73,0	69,3	61,4	64,2	62,8	54,2	56,7	55,4
NC-CNN	-	-	-	-	-	71,3	-	-	-	-	-	-
$BILSTM_a$	80,1	69,4	74,3	81,6	62,3	70,6	-	-	-	-	-	-
Hybrid	80,8	71,5	75,9	84,6	64,9	73,4	-	-	-	-	-	-
PSL global	-	-	71,7	75,3	64,4	69,4	-	-	-	-	-	-
$DMCNN_{DS}$	79,7	69,6	74,3	75,7	66,0	70,5	71,4	56,9	63,3	62,8	50,1	55,7
ATT	n/a	n/a	n/a	78,0	66,3	71,7	-	-	-	-	-	-
ATT_{DS}	n/a	n/a	n/a	78,0	66,3	71,9	-	-	-	-	-	-
$BILSTM_b$	-	-	-	76,1	63,5	69,3	-	-	-	-	-	-
$BILSTM_{d2v}$	-	-	-	77,2	64,9	70,5	-	-	-	-	-	-
GMLATT	80,9	68,1	74,1	78,9	66,9	72,4	-	-	-	-	-	-
$BILSTM_{re}$	73,7	78,5	76,1	71,5	76,3	73,9	-	-	-	-	-	-
graphCNN	-	-	-	77,9	68,8	73,1	-	-	-	-	-	-
BIGRU	-	-	-	66,2	72,3	69,1	-	-	-	-	-	-
DEEB	-	-	-	72,3	75,8	74,0	-	-	-	-	-	-
DAG-GRU	-	-	-	-	-	69,2	-	-	-	-	-	-
JMEE	-	-	-	74,1	69,8	71,9	71,3	64,5	67,7	66,2	52,8	58,7
SELF	75,3	78,8	77,0	71,3	74,7	73,0	-	-	-	-	-	-

Tableau 2. *Résultats des modèles présentés sur le jeu de test ACE 2005 (p : précision, r : rappel, f : f1-mesure)*

Concernant les extensions à l'approche locale, l'augmentation de données proposée par Chen *et al.* (2017) permet de gagner 1,4 point pour l'architecture DMCNN, passant de 69,1 à 70,5. Ce gain est toutefois plus marginal, voir non significatif pour le modèle ATT, ne passant que de 71,7 à 71,9. On peut supposer que l'augmentation de données en volume permet surtout d'exposer le modèle à un plus grand vocabulaire de déclencheurs, ce qui augmente nécessairement les performances des premiers modèles locaux assez centrés sur ces derniers. À l'inverse, le modèle ATT exploite déjà un contexte plus large grâce à l'attention supervisée centrée sur les arguments et est donc probablement moins sensible à ce problème.

L'évaluation de l'intérêt de l'approche jointe n'est pour sa part pas toujours facile car elle n'est pas nécessairement conçue comme l'extension d'une approche existante. On peut toutefois comparer le modèle JointBIGRU adoptant une approche jointe et le modèle DMCNN séquentiel : les performances des deux modèles en prédiction de déclencheurs sont comparables tandis que le modèle JointBIGRU obtient des résultats similaires aux autres modèles récurrents. L'apprentissage joint ne semble donc pas bénéficier à l'extraction des déclencheurs. En revanche, on observe un gain important pour la prédiction des arguments.

Les approches globales semblent offrir un intérêt indéniable : les approches crossSents, crossDocs et crossEvents reposent toutes sur les prédictions du modèle sentRules qu'elles ne font que modifier. On observe des gains importants à la fois pour les déclencheurs et les arguments, jusqu'à 19 points pour les premiers et 8 points pour les seconds pour le modèle crossEvents. Par ailleurs, contrairement aux gains de l'augmentation de données, ces gains ne disparaissent pas pour des modèles plus performants : l'emploi d'un plongement de documents octroie au modèle $\text{BILSTM}_{\text{d2v}}$ un point de plus tandis que le plongement spécifique de DEEB conduit à un gain de 5 points, DEEB se rapprochant ainsi du niveau d'un annotateur humain (Hong *et al.*, 2011). Ceci s'explique par la plus grande sophistication du modèle hiérarchique de document utilisé qui bénéficie grandement de l'attention supervisée durant l'apprentissage ainsi que d'une représentation spécifique à chaque exemple d'apprentissage, contrairement au modèle précédent.

L'analyse comparative des tendances se dégageant de l'état de l'art que nous proposons doit néanmoins être considérée avec prudence. En effet, les modèles reposant sur des architectures neuronales sont dépendants de processus aléatoires. Or, à l'exception de DAG-GRU, les articles ne fournissent qu'une seule valeur pour les performances du modèle alors qu'il faudrait reproduire l'expérience plusieurs fois et donner des performances moyennes (Reimers et Gurevych, 2017). Orr *et al.* (2018) reproduisent aussi fidèlement que possible différents modèles de l'état de l'art et réalisent une analyse empirique rigoureuse synthétisée par le tableau 3. Il en ressort que l'écart entre performances moyenne et maximale est souvent plus important que les gains revendiqués dans les articles. De plus, les résultats maximaux de certains modèles ne sont pas du tout comparables à ceux rapportés. Ces différences peuvent s'expliquer par des configurations très sensibles à l'initialisation et aux valeurs des hyperparamètres ainsi que par l'influence notable des prétraitements sur les résultats finaux.

modèle	moy.	max.	std	publié
DAG-GRU	69,2	71,1	0,91	-
jointGRU	68,0	69,4	0,86	69,3
Hybrid	66,4	68,1	1,32	73,4
JMEE	65,2	66,8	0,94	71,9
CNN	64,7	67,2	1,38	67,6

Tableau 3. *Détection de déclencheurs : moyenne pour 20 tests (Orr* et al.*, 2018)*

7. Conclusion

L'état de l'art de l'extraction d'événements a rapidement évolué ces dernières années grâce au développement des méthodes de construction de plongements lexicaux et aux architectures neuronales. Celles-ci permettent une meilleure prise en compte de la grande variété d'expression revêtue par les événements. Les modélisations locales séquentielles classiques ne permettent cependant pas de désambiguïser l'intégralité des mentions d'événements. Pour pallier ce problème au niveau local, les modélisations exploitant les dépendances syntaxiques ou des mécanismes d'attention montrent des résultats intéressants pour réaliser une interprétation plus fine du contexte local. Les approches jointes ne semblent pas permettre d'améliorer la détection d'événements mais améliorent la détection des arguments en exploitant l'interdépendance entre ces deux tâches. Indépendamment de cette meilleure prise en compte du contexte local, l'inférence globale permet de résoudre un certain nombre d'ambiguïtés insolubles au niveau local. Cependant, il est impossible de véritablement conclure sur l'ensemble de ces tendances, les différents modèles de l'état de l'art souffrant d'un problème de reproductibilité lié à la complexité des prétraitements propres à chaque équipe et à l'absence de prise en compte de l'influence de l'initialisation aléatoire compte tenu de la taille des jeux de données.

Enfin, la grande majorité des modèles ne s'évaluent que sur le jeu de données ACE 2005, amplifiant la sensibilité des résultats aux biais spécifiques de ces données, un problème déjà identifié et particulièrement étudié par la communauté de la vision par ordinateur (Tommasi *et al.*, 2017). À l'avenir, il semble ainsi nécessaire de s'assurer de la robustesse des architectures proposées sur plusieurs jeux de données. De plus, l'application des méthodes de *transfer learning* à l'extraction d'événements, jusqu'alors anecdotique (Bronstein *et al.*, 2015), devrait certainement devenir de plus en plus prégnante. D'une part, la complexification croissante des architectures restreint leur application à des domaines suffisamment dotés en données annotées. D'autre part, tout comme la mise à disposition de modèles neuronaux préentraînés sur la reconnaissance d'objets a donné lieu à l'apparition de nombreuses méthodes de transfert et d'adaptation au domaine vers d'autres tâches visuelles, l'apparition récente de modèles de langues particulièrement imposants (Devlin *et al.*, 2019 ; Peters *et al.*, 2018), tant par leurs performances que leur taille ou le volume de données d'entraînement, va certainement produire une dynamique similaire dans le cadre textuel.

8. Bibliographie

Abend O., Rappoport A., « The State of the Art in Semantic Representation », *55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, p. 77-89, 2017.

Bahdanau D., Cho K., Bengio Y., « Neural machine translation by jointly learning to align and translate », *ICLR 2015*, 2015.

Baker C. F., Fillmore C. J., Lowe J. B., « The Berkeley FrameNet Project », *ACL-COLING'98*, p. 86-90, 1998.

Bengio Y., Ducharme R., Vincent P., Jauvin C., « A Neural Probabilistic Language Model », *Journal of Machine Learning Research*, vol. 3, p. 1137-1155, Feb, 2003.

Blei D. M., Ng A. Y., Jordan M. I., « Latent dirichlet allocation », *Journal of machine Learning research*, vol. 3, p. 993-1022, 2003.

Bollacker K., Evans C., Paritosh P., Sturge T., Taylor J., « Freebase : a collaboratively created graph database for structuring human knowledge », *SIGMOD'08*, p. 1247-1250, 2008.

Bronstein O., Dagan I., Li Q., Ji H., Frank A., « Seed-Based Event Trigger Labeling : How far can event descriptions get us ? », *ACL-IJCNLP 2015*, p. 372-376, 2015.

Chen C., Ng V., « Joint Modeling for Chinese Event Extraction with Rich Linguistic Features », *COLING 2012*, p. 529-544, 2012.

Chen Y., Liu S., Zhang X., Liu K., Zhao J., « Automatically Labeled Data Generation for Large Scale Event Extraction », *ACL 2017*, p. 409-419, 2017.

Chen Y., Xu L., Liu K., Zeng D., Zhao J., « Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks », *ACL-IJCNLP 2015*, p. 167-176, 2015.

Cho K., van Merrienboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y., « Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation », *EMNLP 2014*, p. 1724-1734, 2014.

Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., Kuksa P., « Natural Language Processing (Almost) from Scratch », *Journal of Machine Learning Research*, vol. 12, p. 2493-2537, 2011.

Devlin J., Chang M.-W., Lee K., Toutanova K., « BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding », *NAACL-HLT 2019*, p. 4171-4186, 2019.

Doddington G., Mitchell A., Przybocki M., Ramshaw L., Strassel S., Weischedel R., « The Automatic Content Extraction (ACE) Program – Tasks, Data, and Evaluation », *4^{th} Conference on Language Resources and Evaluation (LREC 2004)*, p. 837-840, 2004.

Duan S., He R., Zhao W., « Exploiting Document Level Information to Improve Event Detection via Recurrent Neural Networks », *IJCNLP 2017*, p. 352-361, 2017.

Dumais S. T., Furnas G. W., Landauer T. K., Deerwester S., Harshman R., « Using Latent Semantic Analysis to Improve Access to Textual Information », *SIGCHI Conference on Human Factors in Computing Systems (CHI'88)*, p. 281-285, 1988.

Feng X., Huang L., Tang D., Ji H., Qin B., Liu T., « A Language-Independent Neural Network for Event Detection », *ACL 2016*, p. 66-71, 2016.

Getman J., Ellis J., Strassel S., Song Z., Tracey J., « Laying the Groundwork for Knowledge Base Population : Nine Years of Linguistic Resources for TAC KBP », *LREC 2018*, 2018.

Grishman R., Sundheim B., « Message Understanding Conference- 6 : A Brief History », *16^{th} International Conference on Computational Linguistics (COLING 1996)*, p. 466-471, 1996.

Grishman R., Westbrook D., Meyers A., « NYU's English ACE 2005 System Description », *ACE*, 2005.

Harris Z. S., « Distributional Structure », *Word*, vol. 10, nº 2-3, p. 146-162, 1954.

Hobbs J. R., Appelt D. E., Bear J., Israel D., Kameyama M., Stickel M., Tyson M., « FASTUS : A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text », *Finite-State Language Processing*, p. 383-406, 1997.

Hochreiter S., Schmidhuber J., « Long Short-Term Memory », *Neural Computation*, vol. 9, nº 9, p. 1735-1780, 1997.

Hong Y., Zhang J., Ma B., Yao J., Zhou G., Zhu Q., « Using Cross-Entity Inference to Improve Event Extraction », *ACL-HLT 2011*, p. 1127-1136, 2011.

Hong Y., Zhou W., Zhang J., Zhu Q., Zhou G., « Self-Regulation : Employing a Generative Adversarial Network to Improve Event Detection », *ACL 2018*, p. 515-526, 2018.

Jean-Louis L., Besançon R., Ferret O., « Text Segmentation and Graph-based Method for Template Filling in Information Extraction », *IJCNLP 2011*, p. 723-731, 2011.

Ji H., Grishman R., « Refining Event Extraction through Cross-Document Inference », *ACL 2008*, p. 254-262, 2008.

Kodelja D., Besançon R., Ferret O., « Exploiting a More Global Context for Event Detection Through Bootstrapping », *ECIR 2019*, p. 763-770, 2019.

Kodelja D., Besançon R., Ferret O., « Représentations et modèles en extraction d'événements supervisée », *RJCIA 2017*, 2017.

Le Q., Mikolov T., « Distributed Representations of Sentences and Documents », *31st International Conference on Machine Learning (ICML 2014)*, p. 1188-1196, 2014.

Li Q., Ji H., Huang L., « Joint Event Extraction via Structured Prediction with Global Features. », *ACL 2013*, p. 73-82, 2013.

Liao S., Grishman R., « Using Document Level Cross-Event Inference to Improve Event Extraction », *ACL 2010*, p. 789-797, 2010.

Liu J., Chen Y., Liu K., Zhao J., « Event Detection via Gated Multilingual Attention Mechanism », *32nd AAAI Conference on Artificial Intelligence*, 2018.

Liu S., Chen Y., He S., Liu K., Zhao J., « Leveraging FrameNet to Improve Automatic Event Detection », *ACL 2016*, p. 2134-2143, 2016a.

Liu S., Chen Y., Liu K., Zhao J., « Exploiting Argument Information to Improve Event Detection via Supervised Attention Mechanisms », *ACL 2017*, p. 1789-1798, 2017.

Liu S., Liu K., He S., Zhao J., « A Probabilistic Soft Logic Based Approach to Exploiting Latent and Global Information in Event Classification. », *AAAI 2016*, 2016b.

Lytinen S. L., Gershman A., « ATRANS Automatic Processing of Money Transfer Messages », *AAAI 1986*, p. 1089-1093, 1986.

McCallum A. K., Nigam K., Rennie J., Seymore K., « Automating the Construction of Internet Portals with Machine Learning », *Information Retrieval*, vol. 3, nº 2, p. 127-163, 2000.

Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., « Distributed Representations of Words and Phrases and Their Compositionality », *NIPS 2013*, p. 3111-3119, 2013.

Miller G. A., « WordNet : A Lexical Database for English », *Communications of the ACM*, vol. 38, nº 11, p. 39-41, 1995.

Mitamura T., Yamakawa Y., Holm S., Song Z., Bies A., Kulick S., Strassel S., « Event Nugget Annotation : Processes and Issues », *3rd Workshop on EVENTS*, p. 66-76, 2015.

Nguyen T. H., Cho K., Grishman R., « Joint Event Extraction via Recurrent Neural Networks », *NAACL HLT 2016*, p. 300-309, 2016a.

Nguyen T. H., Grishman R., « Event Detection and Domain Adaptation with Convolutional Neural Networks », *ACL-IJCNLP 2015*, p. 365-371, 2015.

Nguyen T. H., Grishman R., « Modeling Skip-Grams for Event Detection with Convolutional Neural Networks », *EMNLP 2016*, p. 886-891, 2016.

Nguyen T. H., Grishman R., « Graph Convolutional Networks with Argument-Aware Pooling for Event Detection », *32nd AAAI Conference on Artificial Intelligence*, p. 5900-5907, 2018.

Nguyen T. H., Grishman R., Meyers A., « New York University 2016 System for KBP Event Nugget : A Deep Learning Approach », *6th Text Analysis Conference*, 2016b.

Orr J. W., Tadepalli P., Fern X., « Event Detection with Neural Networks : A Rigorous Empirical Evaluation », *EMNLP 2018*, p. 999-1004, 2018.

Peters M., Neumann M., Iyyer M., Gardner M., Clark C., Lee K., Zettlemoyer L., « Deep Contextualized Word Representations », *NAACL HLT 2018*, p. 2227-2237, 2018.

Reimers N., Gurevych I., « Reporting Score Distributions Makes a Difference : Performance Study of LSTM-networks for Sequence Tagging », *EMNLP 2017*, p. 338-348, 2017.

Riloff E., « Automatically Constructing a Dictionary for Information Extraction Tasks », *AAAI 1993*, p. 811-816, 1993.

Sha L., Qian F., Chang B., Sui Z., « Jointly Extracting Event Triggers and Arguments by Dependency-Bridge RNN and Tensor-Based Argument Interaction », *32nd AAAI Conference on Artificial Intelligence*, 2018.

Stevenson M., « Fact Distribution in Information Extraction », *Language Resources and Evaluation*, vol. 40, n° 2, p. 183-201, 2006.

Tamaazousti Y., Borgne H. L., Hudelot C., Seddik M. E. A., Tamaazousti M., « Learning More Universal Representations for Transfer-Learning », *TPAMI*, 2019.

Tommasi T., Patricia N., Caputo B., Tuytelaars T., « A deeper look at dataset bias », *Domain Adaptation in Computer Vision Applications*, p. 37-55, 2017.

Turian J., Ratinov L., Bengio Y., « Word Representations : A Simple and General Method for Semi-Supervised Learning », *ACL 2010*, p. 384-394, 2010.

Yang B., Mitchell T. M., « Joint Extraction of Events and Entities within a Document Context », *NAACL HLT 2016*, p. 289-299, 2016.

Yangarber R., Jokipii L., « Redundancy-Based Correction of Automatically Extracted Facts », *EMNLP 2005*, p. 57-64, 2005.

Zeng D., Liu K., Chen Y., Zhao J., « Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. », *EMNLP 2015*, p. 1753-1762, 2015.

Zhang J., Zhou W., Hong Y., Yao J., Zhang M., « Using Entity Relation to Improve Event Detection via Attention Mechanism », *NLPCC 2018*, p. 171-183, 2018.

Zhao Y., Jin X., Wang Y., Cheng X., « Document Embedding Enhanced Event Detection with Hierarchical and Supervised Attention », *ACL 2018*, p. 414-419, 2018.

Zhou G., Su J., Zhang J., Zhang M., « Exploring Various Knowledge in Relation Extraction », *43rd Annual Meeting on Association for Computational Linguistics*, p. 427-434, 2005.