Prise en compte du contexte inter-phrastique pour l'extraction d'événements supervisée

Dorian Kodelja Bonan

université
PARIS-SACLAY

Prise en compte du contexte inter-phrastique pour l'extraction d'événements supervisée

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580, Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique
Unité de recherche: Université Paris-Saclay, CEA, Institut LIST, 91191, Gif-sur-Yvette, France.
Référent: : Faculté des sciences

Thèse présentée et soutenue à Palaiseau, le 17 Janvier 2020, par

Dorian KODELJA

Composition du jury:

Anne Vilnat Professeure, Université Paris-Saclay (LIMSI)	Présidente
Sylvie Calabretto Professeure, INSA Lyon (LIRIS)	Rapportrice & Examinatrice
Gaël Dias Professeur, Université Caen Normandie (GREYC)	Rapporteur & Examinateur
Benoit Favre Maître de Conférences, Université Aix Marseille (LIS)	Examinateur
Olivier Ferret Ingénieur chercheur, CEA LIST	Directeur de thèse
Romaric Besançon Ingénieur chercheur, CEA LIST	Coencadrant

Thèse de doctorat

NNT: 2020UPASS005

Table des matières

Table des figures

Liste des tableaux

Introduction

« Simultanément s'accrut la possibilité d'accumuler des connaissances, en particulier des connaissances abstraites, parce que l'écriture modifiait la nature de la communication en l'étendant au-delà du simple contact personnel et transformait les conditions de stockage de l'information. Ainsi fut rendu accessible à ceux qui savaient lire un champ intellectuel plus étendu. Le problème de la mémorisation cessa de dominer la vie intellectuelle ; l'esprit humain put s'appliquer à l'étude d'un texte statique [...], ce qui permit à l'homme de prendre du recul par rapport à sa création et de l'examiner de manière plus abstraite, plus générale, plus rationnelle. »

Jack Goody

L'écriture est un moyen de communication permettant, à travers un ensemble de contraintes additionnelles à celles de l'oral, de garantir la pérennité des informations transmises. Cette garantie de durabilité et d'unité de l'information justifie le rôle fondamental que l'écriture a joué dans le développement des sciences, la codification des lois, la consignation des faits historiques et plus généralement l'accumulation et la diffusion de connaissances. L'essor récent des sciences informatiques et des technologies de télécommunication, couplé à la tertiarisation actuelle de l'économie permise par l'automatisation industrielle a ainsi fait apparaître ou croître nombre d'activités reposant fortement sur la production et l'exploitation d'informations sous forme textuelle. Que ce soit dans la presse, la recherche académique, l'industrie ou le droit, la quantité de données textuelles et le besoin d'accéder à l'information contenue dans ces données est en pleine croissance. Ce volume de données toujours plus conséquent est donc à la fois une formidable richesse à exploiter et un important défi technique pour les scientifiques.

Les applications pratiques sont en effet nombreuses dans les domaines nécessitant l'analyse et la synthèse de grands volumes de données, telle que la veille informationnelle (technologique, juridique, commerciale), l'analyse de jurisprudences ou l'assistance au diagnostic médical. Cependant, afin de pouvoir analyser ces informations et les mettre en relation, il est nécessaire de pouvoir en extraire une version exploitable et donc structurée. La difficulté de cette extraction repose dans la nature non structurée du texte. Une même information peut être exprimée de très nombreuses manières, avec de grandes variations de style et de vocabulaire. À l'inverse, l'ambiguïté de la langue implique qu'une même portion de texte peut être interprétée de différentes manières et désigner des informations différentes. Ces difficultés sont par ailleurs amplifiées par la pluralité des sources d'informations : outre les domaines de spécialité (science, médecine, droit, économie) aux normes très spécifiques, les documents du domaine général (presse, sites d'actualité, bulletins télévisés, blogs et forums) présentent également d'importantes variations des conventions d'écriture. Il est donc nécessaire de pouvoir distinguer l'information pertinente dans les textes de façon propre à ces différents types de discours et de pouvoir l'interpréter correctement.

Extraction d'événements

Le domaine de l'Extraction d'Information (IE) répond à ce besoin et consiste dans sa forme la plus complète à extraire des termes puis à les mettre en relation avec des structures de données prédéfinies, assimilables à des formulaires, pour permettre de les stocker dans des bases de données pour différents usages futurs. Les différents types de structures de données permettent ainsi de distinguer les principales tâches de ce domaine de recherche : l'Extraction d'Entités Nommées (NER), l'Extraction de Relations (RE) et l'Extraction d'Événements (EE).

L'extraction d'événements consiste en l'identification d'instances d'événements, de type préalablement définis, ainsi que des entités tenant les différents rôles les caractérisant, tels que leur lieu, leur date et les différents acteurs de ces événements. Nous illustrons en Figure A un cas typique. Cette tâche peut ainsi être vue comme l'extraction d'une rela-

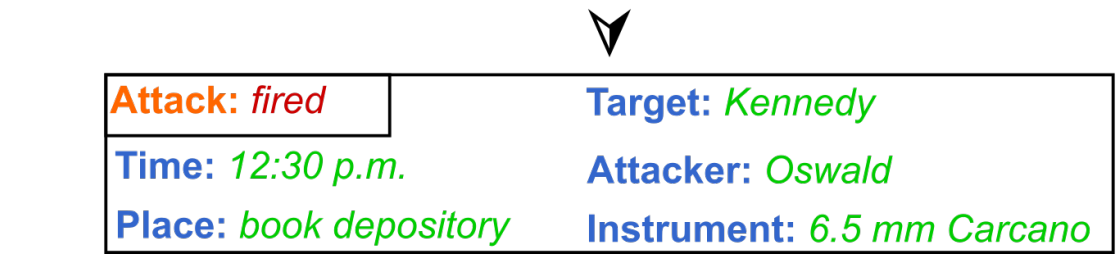

FIGURE A – À partir de la phrase en exemple, on peut identifier le mot "*fired*" comme déclencheur d'un événement de type *Attack*. Cet événement est associé à un formulaire, caractérisé par différents rôles (en bleu). L'extraction des arguments consiste alors à identifier parmi les entités de la phrase (en vert), celles remplissant un rôle au sein de l'événement.

tion n-aire entre différentes entités, chacune ayant un rôle spécifique dans l'événement. Les premiers jeux de données et, en réponse, les premiers modèles, se focalisaient sur le résultat final de l'extraction, c'est-à-dire les formulaires remplis à l'échelle du document. L'accent n'était donc pas mis sur l'ancrage spécifique des différentes entités dans le texte. À l'inverse, les jeux de données ultérieurs ont introduit le concept de déclencheur événementiel, désignant le ou les mots indiquant le plus clairement la présence d'un événement dans une phrase. L'extraction est alors envisagée en deux temps : l'identification et la classification du déclencheur en l'un des types d'événements, appelée Détection d'Événements (*Event Detection*, ou ED) puis l'identification et la classification, parmi les entités de la phrase, de celles tenant un rôle dans l'événement. Cette modélisation met ainsi l'accent sur l'extraction d'événements au niveau intra-phrastique.

Problématique

L'extraction d'événements est une tâche complexe nécessitant la détection d'interactions fines entre des entités et des déclencheurs au sein de textes non structurés. Il est donc nécessaire de pouvoir exploiter au mieux les informations contenues dans le document. Dans cette thèse, nous faisons référence à l'ensemble de ces informations en tant que contexte. Plus précisément, lors de l'application d'un modèle à une phrase, les infor-

mations contenues dans cette phrase constituent le contexte intra-phrastique, tandis que le contexte inter-phrastique désigne les informations présentes dans le reste du document. Or, comme nous venons de le voir, la structuration actuelle de la tâche encourage une focalisation intra-phrastique des modèles. Cette incitation est compréhensible puisque ce contexte intra-phrastique est suffisant dans la majorité des cas. Cependant, l'exploitation de ce contexte repose sur deux prérequis : la présence d'informations suffisantes pour détecter la présence d'un événement et la capacité du modèle à accéder à cette information. Dans le cas du second prérequis, les modèles actuels sont encore perfectibles concernant l'exploitation des dépendances longues au sein de la phrase. Si l'on peut considérer que l'amélioration des performances de ces modèles intra-phrastiques permettra de répondre, à terme, à ce second prérequis, il reste néanmoins indispensable de prendre en compte le contexte inter-phrastique pour répondre au premier.

Pour illustrer ces situations, nous présentons deux phrases issues d'un même article de presse.

— « A police officer in Louisiana has been fired after he published a Facebook post insinuating Rep. Alexandria Ocasio-Cortez (D-NY) should be shot. »

— « Chief Arthur Lawson of the Gretna Police Department announced the terminations. »

Le mot *"fired"* dans la première phrase fait référence à un licenciement mais peut également évoquer l'utilisation d'une arme à feu. Or, ces deux événements font partie des taxonomies d'événements utilisées. Un modèle intra-phrastique pourrait donc prédire le mauvais type d'événement. De plus, la présence du mot "*shot*" dans la phrase renforce l'ambiguïté. Dans ce cas, la présence du terme "*terminations*" dans une autre phrase renforce l'interprétation de "fired" en tant que déclencheur d'un événement de type licenciement.

Cet exemple illustre l'intérêt de l'exploitation du contexte inter-phrastique. Notre problématique consiste alors à étudier les manières de réaliser une telle prise en compte.

La grande majorité des approches actuelles les plus performantes reposent sur des architectures neuronales. Une première déclinaison de notre problématique consiste donc

à étudier les possibilités de prise en compte du contexte inter-phrastique par ces architectures. Ces modèles sont généralement appliqués à des séquences de mots au sein du contexte intra-phrastique. Au-delà de la difficulté à exploiter des dépendances à l'échelle d'un document, la taille des séquences que ces architectures peuvent exploiter est limitée en pratique par les ressources de calculs importantes qu'elles nécessitent. Il n'est donc pas envisageable de simplement considérer l'ensemble du document comme une seule séquence sur laquelle appliquer le modèle. Dès lors, le contexte inter-phrastique doit être considéré différemment du contexte intra-phrastique. Nous chercherons donc à extraire une représentation compacte du contexte inter-phrastique puis à l'intégrer au modèle en complément de la modélisation fine du contexte intra-phrastique. Nous faisons l'hypothèse, confirmée dans ces travaux, qu'il est souhaitable que cette représentation du contexte inter-phrastique soit spécifique à la tâche et à l'exemple à prédire, par opposition à une représentation générique et unique des documents.

L'utilisation du contexte inter-phrastique sous forme d'une représentation permet ainsi d'exploiter la présence d'informations distantes pour résoudre des ambiguïtés au niveau intra-phrastique. Toutefois, cette solution présuppose qu'il est possible de résoudre séquentiellement et indépendamment chaque ambiguïté intra-phrastique.

Une manière plus directe de considérer le contexte inter-phrastique, qui s'affranchit de ce présupposé, consiste à modéliser directement les interdépendances entre les différents événements d'un document et à réaliser une prédiction globale afin de déterminer la meilleure assignation de l'ensemble des événements à l'échelle de ce document. Comme nous l'avons évoqué précédemment, cette approche n'est pas modélisable à l'aide des approches neuronales ni, par extension, par des méthodes reposant sur des classifieurs. Cependant, compte tenu de la difficulté de la tâche d'extraction d'événements et de la taille relativement élevée des documents considérés, l'application directe d'un modèle global aux tokens d'un document n'est pas non plus envisageable. Nous proposons donc d'étudier la prédiction globale d'événements comme l'amélioration a posteriori des prédictions d'un modèle intra-phrastique.

Nous explorons donc ces deux axes de prise en compte du contexte global au travers

de plusieurs contributions détaillées dans les prochains chapitres de ce manuscrit dont nous présentons à présent l'organisation.

Structure du manuscrit

Cette thèse est structurée de la manière suivante. Dans le chapitre 1, nous proposons une vue d'ensemble de l'extraction d'événements. Nous y présentons tout d'abord un historique des campagnes d'évaluations et des différents modèles en détection d'événements avant de définir une grille de lecture des différentes architectures neuronales récentes. Nous présentons ensuite une analyse comparative des performances de ces différents modèles sur la campagne d'évaluation ACE 2005.

Le chapitre 2 consiste en une étude approfondie d'un modèle convolutif de l'état de l'art. Ce modèle consiste un réseau de neurones convolutif appliqué à une fenêtre de taille fixe centrée sur la cible de prédiction, la représentation d'entrée de chaque mot étant obtenue par concaténation de différentes représentations vectorielles. Ce modèle apprend ainsi automatiquement à extraire des descripteurs sémantiques permettant de caractériser l'exemple et à lui associer un type d'événements. Nous présentons une analyse détaillée de certains paramètres et nous évaluons ce modèle sur différents jeux de données. Dans un second temps, nous nous intéressons à la taxonomie des événements à travers plusieurs expériences d'apprentissage incrémental et concluons notamment sur la nécessité de traiter l'identification et la classification des événements de manière jointe. Enfin, nous présentons une étude des performances du modèle en fonction de la taille de la fenêtre de contexte considérée et montrons ainsi que le modèle n'est pas en mesure d'exploiter des dépendances longues. Cette constatation nous amène à formuler la problématique de cette thèse, visant à étudier la prise en compte du contexte inter-phrastique.

Dans le chapitre 3, nous détaillons une première méthode de prise en compte du contexte inter-phrastique. Cette méthode consiste à appliquer un modèle convolutif proche du modèle présenté dans le chapitre 2, puis à agréger les prédictions de ce modèle pour un certain contexte autour du mot cible. Cette étape permet d'obtenir une estimation de la distribution des événements à l'échelle de ce contexte par le modèle local. Cette

représentation est alors intégrée à un autre modèle convolutif, lui permettant de tenir compte de ces informations distantes. Nous considérons trois niveaux d'agrégation : à l'échelle de la phrase cible, à l'échelle de la phrase cible et des phrases précédente et suivante et enfin à l'échelle du document. Nos expériences montrent que l'agrégation au niveau du document est la meilleure. Concernant l'intégration, nous considérons deux niveaux d'intégration : en entrée du modèle ou à la sortie du modèle convolutif avant la couche de prédiction. Les résultats expérimentaux montrent que l'intégration au niveau de la couche de prédiction est la plus pertinente. La comparaison de notre modèle aux modèles de l'état de l'art, ainsi qu'à une représentation générique de document, confirme ainsi la pertinence de cette représentation centrée sur la tâche.

Le chapitre 4 présente l'intégration d'une nouvelle représentation de document à un modèle de convolution de graphe de l'état de l'art. Nous faisons l'hypothèse qu'une représentation de document non seulement centrée sur la tâche comme dans le chapitre précédent, mais également spécifique à chaque exemple est souhaitable. Pour ce faire nous exploitons les coréférences des mentions d'entités de la phrase cible pour identifier les phrases liées au sein du document. Nous appliquons ensuite un modèle récurrent bi-directionnel à ces phrases de contexte puis agrégeons ces représentations en un vecteur unique que nous fournissons en entrée du modèle local. Nous évaluons notre modèle sur plusieurs jeux de données et nous le comparons également à une représentation de document non spécifique à l'exemple et obtenons les meilleures performances pour un modèle simple (par opposition aux modèles d'ensemble), illustrant ainsi la pertinence de notre approche.

Dans le chapitre 5, nous considérons une autre approche de la prise en compte du contexte inter-phrastique. Nous cherchons à modéliser plus directement les interdépendances entre les différentes instances d'événements au sein d'un document afin de réaliser une prédiction jointe. Nous utilisons pour cela le cadre d'apprentissage PSL (*Probabilistic Soft Logic*) qui permet de modéliser de telles interdépendances sous forme de règles en logique du premier ordre. Nous proposons d'améliorer les performances d'un modèle local (intra-phrastique) de l'état de l'art à l'aide de ces règles. Dans un premier temps, nous

nous intéressons à l'amélioration des performances en détection d'événements, puis en extraction d'arguments dans un second temps. Ces deux expériences permettent d'obtenir des gains par rapport au modèle local mais nous nuançons dans un troisième temps les résultats obtenus pour l'extraction d'arguments.

Nous présentons, en conclusion, une synthèse de nos différentes contributions sur la prise en compte du contexte inter-phrastique pour l'extraction d'événements ainsi que plusieurs perspectives de développement des méthodes proposées et des propositions de nouvelles pistes pour des travaux futurs.

Chapitre 1

État de l'Art

Sommaire

1.1 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, source d'information pas ou peu structurée. 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'agence 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 1.2, 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 consi-

dérées 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 1.3 de ce chapitre par présenter les différentes campagnes d'évaluation ayant fortement guidé le développement de cette tâche. Nous donnons ensuite, dans la section 1.4, une vue d'ensemble des différents types de méthodes proposées pour l'extraction d'événements jusqu'à l'émergence des approches neuronales. La Section 1.5 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. L'augmentation de données, notamment via l'utilisation de ressources externes, est l'une d'elles, étudiée à la Section 1.6.1. La Section 1.6.2 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 1.6.3. Les résultats de plusieurs modèles de l'état de l'art appartenant à ces différentes catégories sont présentés puis analysés comparativement à la Section 1.7.

1.2 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, référence pour l'extraction d'événements présentée à la

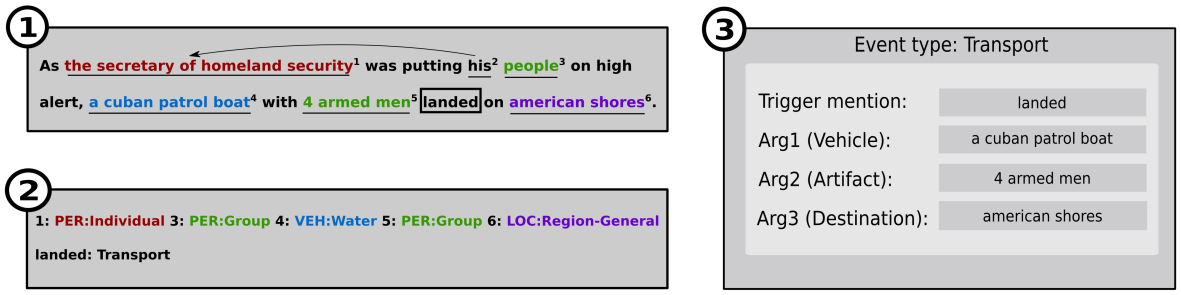

FIGURE 1.2.1 – *La reconnaissance d'entités nommées* identifie les différentes mentions d'entités (soulignées dans le cadre 1) de la phrase et leur type d'entité (cadre 2), auxquelles s'ajoutent les mentions obtenues via les liens de coréférence (cf. flèche). La *détection d'événements* identifie les déclencheurs événementiels 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.

Section 1.7 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 un événement comme une forme de relation n-aire caractérisant une configuration d'entités sans poser de contrainte forte quant à la nature de cette configuration.

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 corres-

pond 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 1.2.1, 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.

1.2.1 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.

1.2.2 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 événementiel, 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 permettent ainsi de lier 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 mettre en évidence les liens entre les déclencheurs faisant référence au même événement, les relations entre déclencheurs pouvant aussi être de nature causale ou temporelle.

1.2.3 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 questionnés. 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 dans 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 intra-phrastique, 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 et nécessite d'intégrer ces différentes extractions en s'appuyant sur les relations de coréférence évoquées précédemment. Comme nous le verrons à la Section 1.6.3, 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 faible 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. De ce fait, l'appariement entre cadre sémantique et événement ne se fait pas toujours facilement et les rares travaux ayant exploité FrameNet (Chen

et al., 2017; Liu *et al.*, 2016a) 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 (Grishman, 2019). 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.

1.3 Campagnes d'évaluation

Le développement de l'extraction d'information à partir de textes a été fortement marqué et guidé par un ensemble de campagnes d'évaluation. Nous présentons ici les campagnes ayant plus particulièrement intégré la notion d'extraction d'événements et qui ont motivé l'apparition et le développement des tâches présentées précédemment.

1.3.1 Message Understanding Conferences

Les campagnes d'évaluation MUC[1] sont une série de conférences organisées par le DARPA (*Defense Advanced Research Projects Agency*) afin de stimuler la recherche en extraction d'information, initialement pour l'analyse de documents militaires. La première

1. `http://itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/proceedings_index.html`

campagne d'évaluation MUC (1987) était relativement peu contrainte, sans critères d'évaluation formels. Elle portait sur des documents traitant de repérages et d'interventions en mer. À partir de la deuxième édition (1989), la tâche fut précisément définie comme le remplissage d'un formulaire. Cette tâche s'est complexifiée au fil des éditions pour atteindre 47 champs différents au sein de 11 formulaires (types d'événements) différents pour la cinquième édition. Le principal tournant dans ce programme fut l'introduction de la précision et du rappel comme métriques d'évaluation lors de MUC-3 (Grishman, 2019). Les résultats obtenus lors de la cinquième édition (1993) furent plutôt satisfaisants mais les systèmes étaient fortement spécialisés. Cette spécialisation implique un lourd travail d'adaptation pour chaque nouveau domaine, loin de l'objectif de conception d'un système générique et universel. Pour répondre à ces attentes, les deux dernières éditions de MUC ont identifié des sous-tâches considérées comme fondamentales et pouvant faire l'objet d'évaluations spécifiques : la reconnaissance d'entités nommées et la résolution de coréférences.

1.3.2 Automatic Content Extraction

Dans la continuité de MUC, les campagnes ACE [2] ont proposé une tâche de détection et de suivi d'entités incluant la reconnaissance d'entités nommées et la résolution de coréférences. L'édition 2004 a introduit la tâche de détection et de caractérisation de relations, qui consistait à extraire des relations étiquetées selon 24 types.

ACE 2005 marque un tournant, avec l'introduction de l'extraction d'événements accompagnée d'un corpus de 599 documents provenant de différentes sources : des dépêches d'agence de presse, des bulletins et débats télévisés, des blogs et groupes de discussion en ligne et enfin, des transcriptions d'échanges téléphoniques. Cette pluralité de sources conduisit à imposer ACE 2005 comme un cadre de référence pour l'extraction d'événements, permettant par ailleurs de tester certaines formes d'adaptation au domaine pour cette tâche (Nguyen et Grishman, 2015a). Suite à (Ji et Grishman, 2008), un découpage s'est imposé, avec 529 documents (3427 déclencheurs) de différentes sources pour l'ap-

2. `http://www.itl.nist.gov/iad/mig/tests/ace/`

prentissage, 40 dépêches (304 déclencheurs) pour le test et 30 documents de différentes sources (350 déclencheurs) pour la validation. La tâche de détection et de caractérisation d'événements couvre 6 types d'événements et 33 sous-types ainsi que 34 rôles pour les arguments. Une des difficultés notables de ce jeu de données est le déséquilibre entre les classes, la classe la plus fréquente, *Attack* apparaissant 1 120 fois tandis que la classe *Pardon* n'apparaît que deux fois. De plus, bien que ce jeu de données soit encore celui de référence aujourd'hui, nous attirons l'attention du lecteur sur le biais du jeu de validation concernant le type de documents[3]. Par ailleurs, le nombre d'exemples relativement faible du jeu de validation rend la recherche d'hyperparamètres plus difficile. Enfin, la présence d'erreurs d'annotations peut également être problématique compte tenu de la faible taille du jeu de test. Certaines équipes utilisant des annotations corrigées quand d'autres ne le font pas, il devient difficile de comparer différentes approches. Plus généralement, compte tenu de la taille du jeu de test, de faibles différences de prétraitement peuvent avoir d'importantes différences sur le jeu de test, rendant la reproduction de travaux difficile. Pour ces différentes raisons, un certain nombre d'articles commencent à proposer des évaluations sur d'autres jeux de données.

1.3.3 Text Analysis Conferences

La campagne TAC[4] (*Text Analysis Conferences*) KBP (Knowledge Base Population) est axée sur une problématique générale de peuplement de base de connaissances et se compose de plusieurs sous-tâches d'extraction d'information : l'*entity linking* pour identifier et lier les entités d'un texte à celles d'une base de connaissances existante ainsi qu'ajouter les entités manquantes à la base ; le *slot-filling* pour extraire des textes des informations factuelles à propos des entités de la base ; enfin, l'extraction d'événements et de relations entre ces événements. Cette dernière tâche est annotée au format Rich ERE, (*Entity, Relation and Event*), un schéma d'annotation très proche du format utilisé par les évaluations ACE, les différences essentielles étant la possibilité d'associer plusieurs évé-

3. "Apparently Ralph and I did not expect the community to follow the data split in our ACL08 paper for almost a decade :-) If we knew it we might have done cross-validation instead." Ji, Heng sur la mailing list TAC à propos du découpage des données.

4. `https://tac.nist.gov/`

nements à un même déclencheur et la présence de déclencheurs multi-tokens. Par ailleurs Rich ERE comporte 38 sous-types, par l'introduction d'un nouveau type et sous-type correspondant (*Manufacture* - *Artifact*) et l'introduction de nouveaux sous-types affinant certains types. Nous fournissons une comparaison des taxonomies ACE et Rich ERE en Figure 1.3.1. L'aspect le plus notoire des campagnes d'évaluations TAC est le volume beaucoup plus important des jeux de données proposés : l'édition 2015 est constituée d'un jeu d'apprentissage contenant 158 documents (6 538 événements) et d'un jeu de test contenant 202 documents (6 438 événements), le jeu de test 2016 contient 169 documents (4 155 événements) et celui de 2017, 167 documents (4 375 événements). Par comparaison avec le jeu de données ACE, les campagnes TAC présentent ainsi 10 à 20 fois plus d'événements, rendant les résultats sur ces jeux de données plus stables et plus aisément reproductibles.

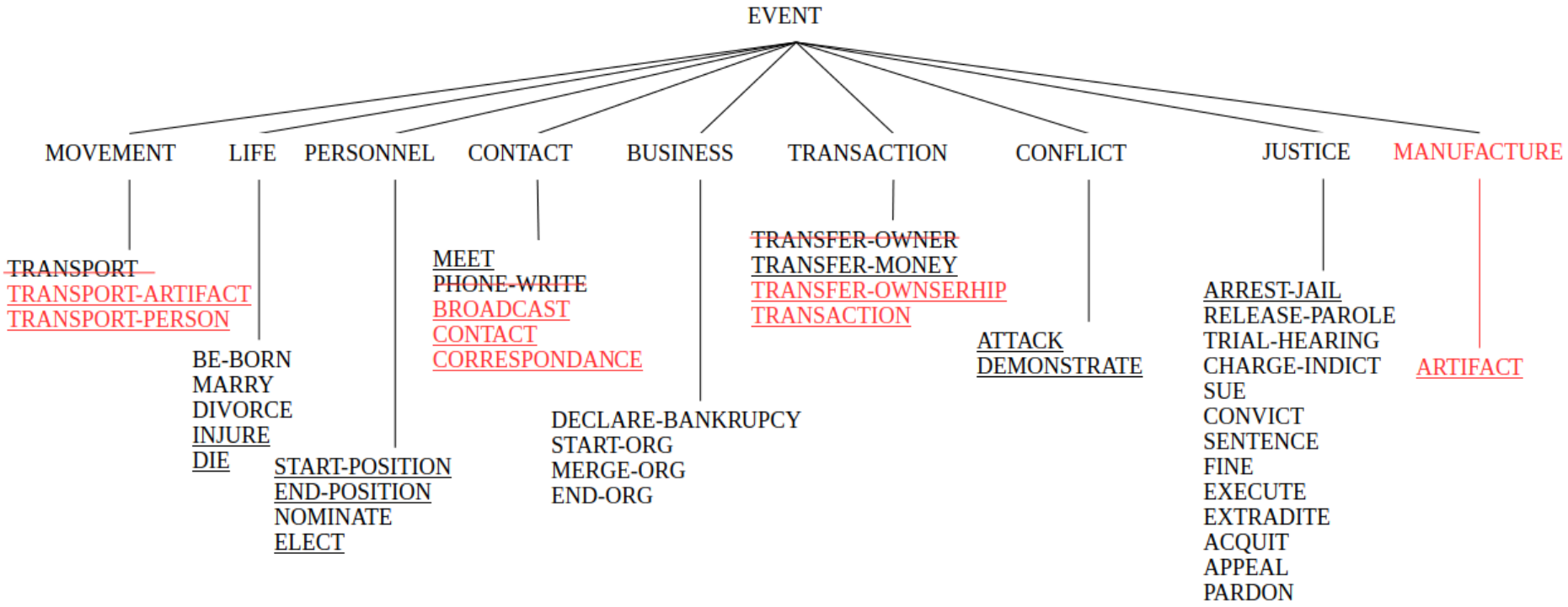

FIGURE 1.3.1 – Présentation des taxonomies d'événements ACE 2005 et Rich ERE : les types et sous-types en noir sont communs aux deux schémas d'annotation. Les termes en rouges sont propres à Rich ERE et ceux barrés de rouge propres à ACE 2005. Les campagnes TAC 2016 et 2017 ne considèrent que les événements soulignés.

1.3.4 Autres campagnes

Il existe également plusieurs campagnes d'évaluation en domaine spécialisé, en particulier dans le domaine biomédical afin d'exploiter les milliers d'articles qui s'y publient chaque jour. Les différentes campagnes BioCreatives [5] et i2b2 [6] ont ainsi fourni des jeux de données annotés sur l'identification de relations entre traitements et maladies ou entre protéines et gènes par exemple.

1.4 Historique

Bien que les approches existantes pour l'extraction d'événements présentent des différences importantes, il est possible de les comparer selon plusieurs axes :

— les systèmes ou les différents modules d'un système peuvent être plus ou moins dépendants du domaine concerné ;

— les modèles peuvent exploiter un degré variable de connaissances linguistiques ;

— la conception des modèles peut nécessiter plus ou moins de données annotées.

Les premiers systèmes d'extraction d'information à partir d'un texte (Hobbs, 1986) se voulaient universels. Ils ne faisaient pas l'hypothèse de l'existence d'un domaine particulier ou d'un type d'information spécifique à extraire. Ces systèmes visaient à réaliser une analyse complète du document (syntaxique, sémantique et pragmatique) afin de comprendre le texte dans son ensemble. Bien que cette approche soit théoriquement pertinente pour "résoudre" l'extraction d'information indépendamment du domaine ou de la tâche cible, ces systèmes étaient trop complexes à développer et nécessitaient trop de ressources [7] et de connaissances à modéliser. De manière générale, la compréhension de texte suppose une analyse en profondeur de l'intégralité des documents. L'implémentation et l'application de ces traitements s'avèrent impossibles, même à l'heure actuelle. En se fixant une tâche moins ambitieuse par l'introduction des notions de domaine et d'information structurée et

5. `http://www.biocreative.org/`
6. `https://www.i2b2.org/NLP/`
7. Les développeurs de TACITUS rapportaient que 36h de calculs étaient nécessaires pour les 100 messages du jeu de test MUC-3 (Hobbs *et al.*, 1997).

spécifique (entités, attributs, relations et événements), l'extraction d'information se différencie de la compréhension de texte par la réduction de la profondeur et de la couverture de l'analyse linguistique nécessaire au fonctionnement d'un système.

1.4.1 Approches à bases de connaissances

Utilisation de motifs lexico-syntaxiques. Ce changement de paradigme pour la conception permet l'apparition, dès la fin des années 1980, des premiers systèmes d'extraction d'information. Ce changement annonce aussi la prédominance des approches intra-phrastiques. En effet, puisque l'objectif n'est plus la compréhension globale du texte mais l'extraction ponctuelle d'informations, de nombreux systèmes ne travaillent dans un premier temps qu'au niveau local. Une phase de consolidation est généralement réalisée à la fin pour fusionner les formulaires construits localement. Ces premiers systèmes, comme ATRANS (Lytinen et Gershman, 1986) ou SCISOR (Rau, 1988), tiraient leur efficacité de règles et d'expressions régulières définies manuellement. De ce fait, ils nécessitaient toujours une conception complexe réalisée spécifiquement par un expert et propre au domaine cible. Ils se caractérisaient par un aspect rigide et monolithique les rendant difficilement adaptables à d'autres langues ou domaines. Le système FASTUS (Hobbs *et al.*, 1997) popularise sur la campagne MUC-3 l'utilisation d'automates à états finis en cascade et plus généralement l'approche séquentielle. Ce système réalise ainsi séquentiellement 5 étapes de reconnaissance de motifs et de *chunking*, la sortie de chaque module étant l'entrée du module suivant. L'intérêt de cette décomposition est l'apparition de la modularité au sein des systèmes d'extraction d'information. Celle-ci permet une modification plus aisée qui facilite le développement et l'adaptation des systèmes. De plus, les trois premières étapes opèrent au niveau linguistique et sont très peu dépendantes du domaine. De ce fait, l'adaptation au domaine ne concerne que les 2 dernières étapes.

Extraction de motifs lexico-syntaxiques. Si les systèmes à base d'automates en cascade marquent un tournant au regard de la lourdeur de la tâche de conception d'un système, cette conception est toujours manuelle et nécessite l'intervention de connaissances expertes à la fois sur le système, la tâche et le domaine. Un nouveau changement de

paradigme intervient avec l'utilisation de méthodes d'extraction de motifs. Le travail nécessaire pour l'adaptation à un nouveau domaine passe ainsi de la conception des règles à l'annotation d'un corpus.

Ces méthodes, à l'instar de RAPIER (Mooney et Califf, 1999) ou d'Autoslog (Riloff, 1993), utilisent différentes représentations des exemples telles que des sacs de mots, un étiquetage en parties du discours ou des arbres syntaxiques. (Grishman *et al.*, 2005) propose une approche séquentielle d'identification des déclencheurs puis des arguments pour finir par la classification du type d'événements. Ce système s'appuie sur l'utilisation de structures syntaxiques et de classifieurs séquentiels et constitue donc aussi un précurseur des approches à base de classifieurs.

1.4.2 Apprentissage de classifieurs

À la différence des systèmes précédents, les systèmes utilisant des classifieurs considèrent la tâche d'extraction d'événement comme une tâche de classification de séquence. Un texte est un ensemble de phrases traitées comme des séquences de *tokens*. La détection d'événement consiste alors à appliquer à chaque élément de la séquence un classifieur entraîné à détecter les déclencheurs et leur type, séquentiellement ou de manière jointe. Il en va de même pour les arguments, généralement prédits parmi les entités nommées détectées en amont.

On distingue au sein de cette famille d'approches deux tendances. La majorité des études utilisent des approches séquentielles en traitant d'abord l'identification et la classification des déclencheurs puis celle des arguments (Ahn, 2006; Chen et Ji, 2009; Grishman *et al.*, 2005). Mais certaines études mettent en œuvre également des approches jointes (Chen et Ng, 2012; Li *et al.*, 2013). Ces méthodes tentent de réduire le problème de propagation des erreurs, symptomatique des approches séquentielles. De plus, elles peuvent ainsi tenir compte de l'interdépendance entre arguments et déclencheurs ou entre détection et caractérisation des déclencheurs ou des arguments. Néanmoins, ces approches se rejoignent sur les types de classifieurs et de représentations utilisés, avec une évolution parallèle à celle de l'extraction de relations. C'est pourquoi nous citerons ici indifféremment

des études portant sur les deux tâches. Les classifieurs sont le plus souvent des machines à vecteurs de support (Hong *et al.*, 2011; Liao et Grishman, 2010; Zhou *et al.*, 2005) ou des classifieurs de type maximum d'entropie (Nguyen et Grishman, 2014; Sun *et al.*, 2011). L'efficacité de ces approches étant particulièrement dépendante de la qualité des représentations choisies, la création de représentations adaptées est essentielle. Les approches à base de classifieurs ont ainsi supprimé l'effort d'élaboration de règles mais l'ont remplacé par un effort d'ingénierie des représentations lui aussi conséquent.

Cependant, une fois des représentations efficaces obtenues, celles-ci s'avèrent assez génériques pour être transposées dans des domaines proches. Zhou *et al.* (2005) introduisent pour l'extraction de relations la plupart des traits (*features*) utilisés dans l'état de l'art. Ces représentations sont produites à plusieurs niveaux : au niveau lexical (sac de mots et tête de mention des déclencheurs, premiers et deuxièmes mots des contextes gauche, milieu, 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)). Sun *et al.* (2011) reprennent ces représentations et complète la représentation lexicale par l'utilisation de bigrammes des mots du contexte central. D'autres informations sémantiques ont été exploitées, telles que les synonymes de WordNet (Li *et al.*, 2013) ou la hiérarchie de prédicats de FrameNet (Li *et al.*, 2014). Il est à noter que le niveau de granularité maximum de ces représentations est généralement le mot bien que des approches descendent au niveau des morphèmes pour l'extraction d'information en chinois (Chen et Ji, 2009). La représentation des mots est généralement de type local ou *one-hot*, c'est-à-dire par un vecteur binaire de taille N où N est la taille du vocabulaire et dont seule la dimension correspondant au mot considéré 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 de capturer convenablement la sémantique des mots ; d'autre part, si les vocabulaires cible et source sont différents, le système ne dispose d'aucune information sur les mots nouveaux. L'utilisation de traits tels que la nature grammaticale d'un mot, sa

lemmatisation, la prise en compte des morphèmes ou de ressources externes telles que des lexiques thématiques ou WordNet permettent partiellement de compenser ces limites. Ces traits sont toutefois d'un intérêt limité, étant trop peu discriminants ou trop dépendants de la disponibilité de nombreuses ressources externes construites manuellement.

1.4.3 Représentations distribuées

Pour répondre aux limitations soulevées précédemment, de nouvelles méthodes ont été proposées pour induire de manière non supervisée des représentations de mots. Ces représentations s'appuient sur l'hypothèse distributionnelle introduite par (Harris, 1954). Cette hypothèse – "You shall know a word by the company it keeps!" (Firth, 1957) – postule que le sens d'un mot peut être déduit des contextes dans lesquels il apparaît et par extension, que des mots apparaissant dans des contextes similaires ont des sens similaires. Il est alors possible d'induire des représentations à partir de la distribution des mots et de leurs contextes au sein de grands corpus de textes. La possibilité de produire de telles représentations à partir de corpus non annotés est particulièrement intéressante pour les modèles d'apprentissage supervisé car il est alors possible d'induire des représentations pour des mots absents du corpus d'apprentissage. Ce faisant, le modèle est capable de mieux généraliser à de nouveaux exemples.

Trois principales familles de représentations de mots peuvent être distinguées dans ce cadre (Turian *et al.*, 2010) : les représentations distributionnelles, les représentations par groupement (*clustering*) et les représentations distribuées ou par plongement lexical (*embeddings*). Nous ne présentons ici que les méthodes par plongement lexical, utilisées par la plupart des approches actuelles. Les représentations par plongement (Bengio *et al.*, 2003; Collobert et Weston, 2008; Mikolov *et al.*, 2013) proposent d'exploiter l'hypothèse distributionnelle en associant à chaque mot un vecteur dense de faible dimension. Ces vecteurs sont dits denses et à faible dimension par opposition à l'encodage one-hot utilisé auparavant : leur nombre de dimensions se réduit généralement à quelques centaines et pour un mot donné, toutes ces dimensions sont actives.

Les méthodes de génération de plongements les plus communes, *CBOW* et *Skip-Gram*,

proposées par (Mikolov *et al.*, 2013) reposent l'une sur la prédiction d'un mot cible à partir de son contexte et l'autre sur la prédiction des mots du contexte à partir du mot cible. D'autres méthodes de plongement ont été définies par la suite, telles que GloVe (Pennington *et al.*, 2014), fastText (Joulin *et al.*, 2017), ELMo (Peters *et al.*, 2018) ou BERT (Devlin *et al.*, 2018) mais les modèles CBOW et Skip-Gram restent les plus usités pour la tâche de détection d'événements, notamment par l'intermédiaire des représentations pré-entraînées fournies par Google en appliquant l'outil *word2vec* à une partie des données GoogleNews.

1.5 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 s'appuyant sur 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 traits 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* :

$$\mathbf{o} = \mathbf{W} \cdot \mathbf{x_{out}} + \mathbf{b} \qquad p(y_j|\mathbf{x}, \theta) = \hat{\mathbf{y}}_j = \frac{\exp^{o_j}}{\sum_{c=1}^{n_c} \exp^{o_c}} \tag{1.1}$$

Les paramètres du modèle peuvent alors être estimés à partir de N exemples d'apprentissage $(\mathbf{x}^i, y^i)$ par optimisation stochastique en utilisant l'entropie croisée (*categorical cross entropy* ou *negative log-likelihood*) comme fonction objectif (*loss function*) :

$$L(\theta) = -\sum_{i=1}^{N} \log p(y^i|\mathbf{x}^i, \theta) \tag{1.2}$$

L'apprentissage conjoint de ces trois parties permet l'émergence de représentations spécifiques adaptées à la tâche de classification considérée (Tamaazousti *et al.*, 2017).

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.

1.5.1 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 type d'entités. Ces représentations complémentaires sont concaténées 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 une forme de prérequis déjà satisfait (Nguyen et Grishman, 2015a). 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* (cf. Section 1.5.3), 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, 2015a) 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.

$$p_w = w - t, \quad -m < p_w < +m \tag{1.3}$$

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.

1.5.2 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 na-

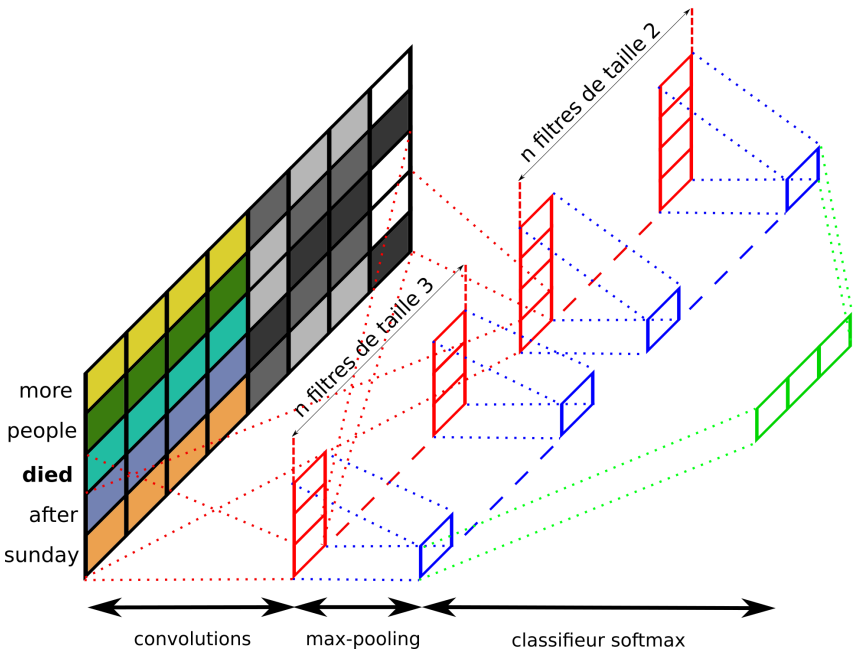

FIGURE 1.5.1 – Schéma de l'architecture d'un CNN.

ture 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 (Nguyen et Grishman, 2015a) 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 $\mathbf{x}_w$ produites précédemment.

Contexte séquentiel court : architectures convolutives. Les réseaux convolutifs, appelés CNN (*Convolutional Neural Network*) et importés de la vision par ordinateur (Chen *et al.*, 2015; Nguyen et Grishman, 2015a), exploitent des cooccurrences locales en utilisant une couche de convolution constituée de plusieurs filtres, ainsi que l'illustre la Figure 1.5.1. Dans le cadre du texte, cette couche ne comprend 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.

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 et génère un vecteur de descripteurs $\mathbf{p_f}$, dont chaque composante est définie par :

$$\mathbf{p_f}_{[i]} = g(\mathbf{x_{i:i+k-1}} \cdot \mathbf{u_f}) \tag{1.4}$$

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 sa position dans le contexte. 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, $\mathbf{s}_{w-1}$:

$$\mathbf{s}_w = \mathrm{RNN}(\mathbf{x}_{1:w}) = \sigma(\mathbf{W_x}\mathbf{x}_w + \mathbf{W_s}\mathbf{s}_{w-1} + \mathbf{b_h}) \tag{1.5}$$

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 rétro-propagation le long de la séquence, il est pratiquement nul pour les premiers mots de la phrase, rendant difficile l'identification

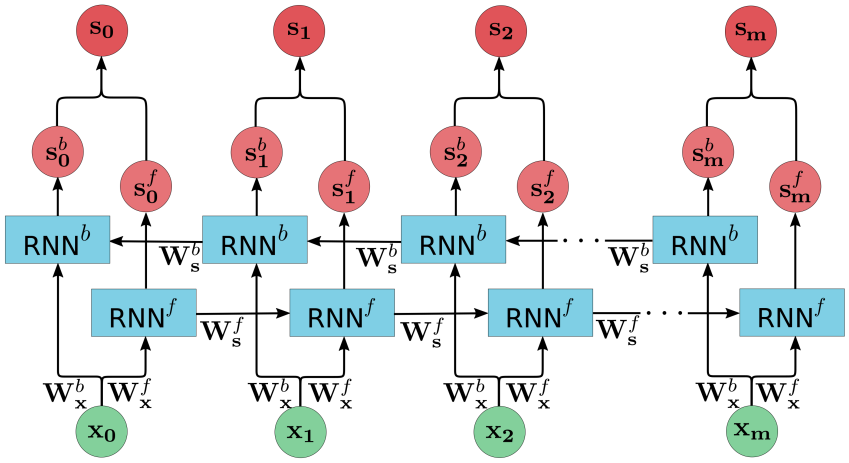

FIGURE 1.5.2 – Schéma de l'architecture d'un RNN bidirectionnel.

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 permettent d'éviter ce phénomène [8].

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 1.5.2, 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.

$$\mathbf{s}_w = [\mathbf{s}^f_\mathbf{w}; \mathbf{s}^b_\mathbf{w}] = [\text{RNN}^f(\mathbf{x}_{1:w}), \text{RNN}^b(\mathbf{x}_{m:w})] \tag{1.6}$$

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 graphe. À l'inverse des approches utilisant

8. Les performances de ces deux modèles sont comparables (Jozefowicz *et al.*, 2015) mais le modèle GRU est plus simple et nécessite d'apprendre moins de paramètres.

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 par exemple un modèle de convolution de graphes opérant exclusivement sur ces représentations sans exploiter le contexte séquentiel. Il est également possible d'utiliser cette information en complément de la représentation de surface : Orr *et al.* (2018) proposent de modifier 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. Cette porte permet à l'état caché du mot courant d'être directement conditionné par les mots avec lesquels il est en relation syntaxique.

1.5.3 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'apprentissage de paramètres supplémentaires. Dans le cas des CNNs, les modèles font appel généralement à une fonction de maxpooling (Nguyen et Grishman, 2015a) conservant la valeur maximale de chaque filtre parmi les valeurs calculées par l'équation 1.4.

$$\mathrm{mP}_{[f]} = \max_{1 \leq w \leq m} \mathbf{p}_{\mathbf{f}[w]} \quad \forall f \in [1, n] \tag{1.7}$$

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 position 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.

$$\mathrm{dmP}_{[f]} = [\max_{1 \leq w \leq t} \mathbf{p_f}_{[w]}; \max_{t \leq w \leq m} \mathbf{p_f}_{[w]}] \quad \forall f \in [1, n] \tag{1.8}$$

Le modèle de convolution de graphe de (Nguyen et Grishman, 2018) utilise l'*entity pooling* appliquant l'opération de *max-pooling* 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 $\mathbf{s_t}$:

$$r_w = f(\mathbf{s_t}, \mathbf{s_w}) \tag{1.9}$$

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 per-

mettant de produire la représentation finale $\mathbf{x_{out}}$ du déclencheur :

$$\mathbf{x_{out}} = \sum_{w=0}^{m} a_w \mathbf{s_w} \qquad a_w = \frac{\exp^{r_w}}{\sum_{j=0}^{m} \exp^{r_j}} \tag{1.10}$$

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.

1.6 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 de manière à 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 humains apparaissant dans la Table 1.7.2 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 des volumes de données bien plus importants que ceux disponibles dans les corpus annotés. 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.

1.6.1 Augmentation de données

Deux types d'augmentation de données ont été mis en œuvre.

Volume. L'augmentation 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 10 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 clustering 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'entité, WordNet permet d'associer à chaque mention d'entité des traits supplémentaires (hyperonymes, synonymes) utilisés ensuite par un algorithme de clustering. 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 fournis en entrée du modèle.

1.6.2 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.*, 2018). 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 classifieur 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.

1.6.3 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 (Jean-Louis *et al.*, 2011; Yangarber et Jokipii, 2005). Ji et Grishman (2008) font ainsi l'hypothèse de la cohérence globale entre mentions et types d'événement (ou de rôle) 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'événement, 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 inter-événement 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, (LDA) (Blei *et al.*, 2003).

Plongement de contexte. 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 modèle local. Selon la même motivation, Kodelja *et al.* (2018) 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 document permettant de produire un plongement du document qui, contrairement à (Duan *et al.*, 2017), est conditionné spécifiquement par la tâche d'extraction d'événements.

1.7 Comparaison

Cadre. Afin d'étudier l'apport des différents choix de modélisation, la Table 1.7.2 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 (60) télévisés, 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 par ailleurs de tester

certaines formes d'adaptation au domaine pour cette tâche (Nguyen et Grishman, 2015a). 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 deux 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 multi-termes. 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 multi-termes. Enfin, ACE 2005 distingue 34 rôles pour les arguments. Les types d'événements peuvent avoir de deux à sept rôles et le plus souvent autour de cinq. Ces rôles sont occupés par des entités au sens de la Section 1.2, qui sont soit générales, comme des personnes ou des organisations, soit plus spécifiques, comme des armes ou des véhicules.

Résultats. Les spécificités des modèles de la Table 1.7.2 sont synthétisées dans la Table 1.7.1 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 nommés 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/structurée de la phrase ;

— **agreg** *(Max/Dyn/ety)P* : (max/dynamic multi/entity)-pooling, *(u/s)Att* : attention (supervisée/non supervisée) ;

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

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, 2015a)	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	eAtt	-	-	-
(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	-	-	-

TABLE 1.7.1 – 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	-	-	-	-	-	-

TABLE 1.7.2 – Résultats des modèles présentés sur le jeu de test ACE 2005 (p : précision, r : rappel, f : F-mesure).

— **joint :** *evt(,ety)* : prédiction jointe déclencheur (entités) et arguments ;

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

Analyse. Comme on peut le constater au niveau de la Table 1.7.1, 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 (Chen *et al.*, 2015; Feng *et al.*, 2016; Nguyen et Grishman, 2015a), 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 pour l'augmentation de données (Chen *et al.*, 2017) vient également mitiger la volonté initiale de dépouillement en termes de dépendances. Sans surprise, l'ajout des entités (CNN_{ets}) permet de mieux résoudre la tâche d'extraction de déclencheurs, avec un apport de 1,4 points 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 toutes des scores entre 69,0 et 69,3 en F-mesure, avec toutefois des profils précision/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.*, 2018). 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, $\text{BILSTM}_{\text{re}}$, 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 distant au niveau intra-phrastique.

Concernant les extensions à l'approche locale, l'augmentation de données proposée par Chen *et al.* (2017) permet de gagner 1,4 points 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 considérant le contexte global 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 dans le cas du modèle crossEvents. On voit d'ailleurs que contrairement

aux gains de l'augmentation de données, ces gains ne disparaissent pas pour des modèles plus performants : l'emploi d'un plongement de document octroie au modèle $\text{BILSTM}_{\text{d2v}}$ un gain de 1 point tandis que le plongement spécifique de DEEB permet d'obtenir un gain de 5 points. 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.

Si nous proposons ici une analyse comparative des tendances se dégageant de l'état de l'art, il est nécessaire de les considérer 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'un seul chiffre pour les performances du modèle alors que la rigueur impose de reproduire l'expérience plusieurs fois et de fournir 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 dont nous présentons les résultats dans la Table 1.7.3. Il en ressort que l'écart entre performance moyenne et maximale est souvent plus important que les gains revendiqués dans les articles. De plus, pour certains modèles, les résultats maximaux ne sont pas du tout comparables à ceux rapportés. Ces différences peuvent notamment s'expliquer par des configurations très sensibles aux variations des hyper-paramètres et à l'initialisation, ainsi qu'à la part prépondérante des prétraitements dans 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

TABLE 1.7.3 – Détection de déclencheurs : moyenne pour 20 tests (Orr *et al.*, 2018).

1.8 Conclusion

Dans ce chapitre, nous avons, dans un premier temps, fait un tour d'horizon de l'extraction d'événements, des premiers systèmes à base de règles aux dernières approches neuronales. Cette présentation a débuté par deux historiques parallèles, l'un concernant les campagnes d'évaluation successives ayant façonné le domaine et l'autre à propos des différents modèles pré-neuronaux. Cette étude nous a permis d'illustrer l'émergence de plusieurs tendances. La transition des systèmes reposant sur des motifs lexico-syntaxiques aux automates en cascade a montré l'intérêt du paradigme séquentiel permettant la réutilisation de modules génériques pour différentes tâches et domaines. Ces automates reposant toujours sur des règles manuelles, l'apparition des systèmes d'extraction de motifs a marqué la transition de la nécessité d'une expertise (pour produire les règles) à celle de données d'entraînement annotées. Par l'introduction du concept de déclencheur, la campagne ACE 2005 a été un tournant important dans la définition même de la tâche d'extraction d'événements. En effet, ce choix de modélisation implique un ancrage spécifique de l'événement dans le texte. Comme nous l'avons vu cet ancrage a induit une focalisation sur le niveau intra-phrastique partagé par la majorité des approches ultérieures. L'émergence conjointe des systèmes à base de classifieurs a mis l'accent sur l'importance du travail d'ingénierie des représentations d'entrée du modèle. Au fil d'une décennie de publications reposant sur des classifieurs (non-neuronaux), les systèmes se sont dotés d'un nombre toujours plus important de traits. À l'exception de quelques traits récurrents (mots, parties du langage, syntagmes), il n'était pas aisé d'identifier l'intérêt propre de chaque trait. Plus globalement, on observe des gains relativement marginaux entre les derniers modèles pré-neuronaux. Suite à la démocratisation des modèles convolutifs en reconnaissance d'image, plusieurs articles ont proposé des systèmes reposant sur des modèles similaires appliqués au texte. Tout comme lors de leur introduction dans la communauté image, les auteurs ont mis en avant la capacité du modèle à exploiter directement les données brutes et produire lui-même les descripteurs sémantiques nécessaires à la résolution de la tâche. Bien que le gain de ces premiers modèles neuronaux par rapport aux modèles non neuronaux contemporains soit modéré, la comparaison de la liste des traits utilisés par Li

et al. (2013) et Chen *et al.* (2015) illustre bien l'intérêt pratique de ce changement de paradigme. Nous avons, dans la suite de ce chapitre, proposé une grille de lecture des nombreuses contributions neuronales récentes. Concernant l'architecture de ces modèles, les modèles récurrents et convolutifs, qui reposent tous deux sur une représentation séquentielle de la phrase, obtiennent des performances proches. Les gains obtenus par les récentes contributions exploitant la structure syntaxique de la phrase en complément de sa forme de surface semblent indiquer que les précédentes architectures sont limitées dans leur capacité de modélisation du contexte intra-phrastique. Le mécanisme d'attention utilisé par plusieurs approches semble également pertinent pour améliorer la prise en compte de ces dépendances locales. Nous avons par ailleurs analysé différentes méthodes permettant d'enrichir le contexte local durant la prédiction : l'augmentation de données, les approches jointes et la prise en compte du contexte global. Parmi celles-ci, la piste la plus prometteuse, bien que relativement négligée, semble l'exploitation du contexte global.

Nous avons enfin identifié plusieurs limites à cette analyse, ces limites n'étant pas dues à des failles inhérentes à notre grille de lecture mais aux conditions d'évaluation des différents modèles. En effet, les différents modèles de l'état de l'art souffrent 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).

De cette analyse, nous pouvons identifier trois perspectives. La première, méthodologique, est la nécessité, à l'avenir, de s'orienter vers des jeux de test plus volumineux tels que les différents jeux de données au format Rich ERE, ainsi que de s'assurer de la robustesse des architectures proposées sur plusieurs jeux de données distincts. Concernant le contexte intra-phrastique, 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 architec-

tures restreint leur application à des domaines suffisamment dotés en données annotées. D'autre part, tout comme la mise à disposition de modèles neuronaux pré-entraînés 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.*, 2018; Peters *et al.*, 2018; Radford *et al.*, 2019), tant par leurs performances que leur taille ou le volume de données d'entraînement, va certainement produire une dynamique similaire dans le cadre textuel. Enfin, concernant les différentes pistes d'amélioration présentées, l'intérêt de la prise en compte du contexte inter-phrastique sous la forme de représentation de document semble bien une piste d'amélioration intéressante. L'application des approches d'inférence globales, jusqu'à présent seulement basés des classifieurs intra-phrastiques classiques, à des modèles neuronaux semblent également une piste intéressante à explorer.

Chapitre 2

Modèle convolutif pour la détection d'événements

Sommaire

Dans ce chapitre, nous proposons d'étudier de façon plus détaillée un modèle convolutif pour la détection d'événements, en présentant ses performances de base et en analysant de façon plus approfondie certains de ses paramètres et de ses propriétés, dans un cadre expérimental donné. Plus précisément, nous présentons en premier lieu l'architecture gé-

nérale du modèle puis les jeux de données utilisés. En accord avec les limitations du jeu de données ACE 2005 présentées dans la section 1.3 du chapitre précédent, nous nous intéresserons aux différentes éditions de la campagne d'évaluation TAC Event Nugget, et plus particulièrement l'édition 2017 à laquelle nous avons participé. Par la suite, nous réalisons des études évaluant la meilleure configuration de type de plongements et de position du dropout. Une fois cette configuration établie, nous évaluons notre modèle sur les jeux de test TAC 2016 et 2017 et obtenons les meilleures performances d'un modèle simple sur ces deux jeux de données. Nous nous intéressons ensuite à la complémentarité de modèles entraînés à différents niveaux de granularité de la taxonomie des événements, démontrant ainsi l'interdépendance forte entre ces différents niveaux et la complexité de la tâche d'extraction d'événements. Enfin, dans un dernier temps, nous présentons une étude de la distance du contexte exploitable par notre modèle montrant la faible taille du contexte intra-phrastique exploitable.

2.1 Réseau de neurones convolutif pour la détection d'événements

L'objectif du modèle est de détecter toutes les mentions d'événements présentes dans un document fourni en entrée. Comme nous l'avons présenté dans la section 1.4.2, les modèles convolutifs font partie des modèles de détection d'événements par classification (neuronale). Nous considérons les phrases comme des séquences de mots auxquels on associe une classe. Cette classification s'appuie sur un prétraitement linguistique des textes. Plus précisément, nous appliquons à chaque document une analyse morphologique (*tokenisation*) et un découpage en phrases. Une analyse syntaxique est ensuite appliquée aux phrases pour extraire à la fois les syntagmes et les dépendances syntaxiques. Dans nos expériences, l'ensemble de ces prétraitements est réalisé à l'aide de l'outil Stanford CoreNLP (Manning *et al.*, 2014). Tout comme Nguyen *et al.* (2016b), dont nous nous inspirons, nous ne considérons que des mentions d'événements mono-token. Ainsi, notre modèle traite la détection d'événements comme une tâche de classification pour chaque

token du document. Le choix de cette approche mono-token est motivé par le fait que les déclencheurs multi-tokens ne représentent respectivement que $1,9\%$ et 3% des déclencheurs événementiels au sein des jeux d'évaluations TAC 2015 et TAC 2016. Nous estimons donc que l'impact négatif de cette modélisation sur les performances est négligeable. En revanche, cette modélisation permet l'introduction de plongements de position dont l'importance est critique pour les performances des modèles convolutifs en extraction d'information (Nguyen et Grishman, 2015b). Puisque nous assignons une classe à l'ensemble des tokens du document, dont une part importante ne sont pas des déclencheurs, nous introduisons une classe supplémentaire, "None", dénotant que le token considéré n'est déclencheur d'aucun des types d'événements considérés.

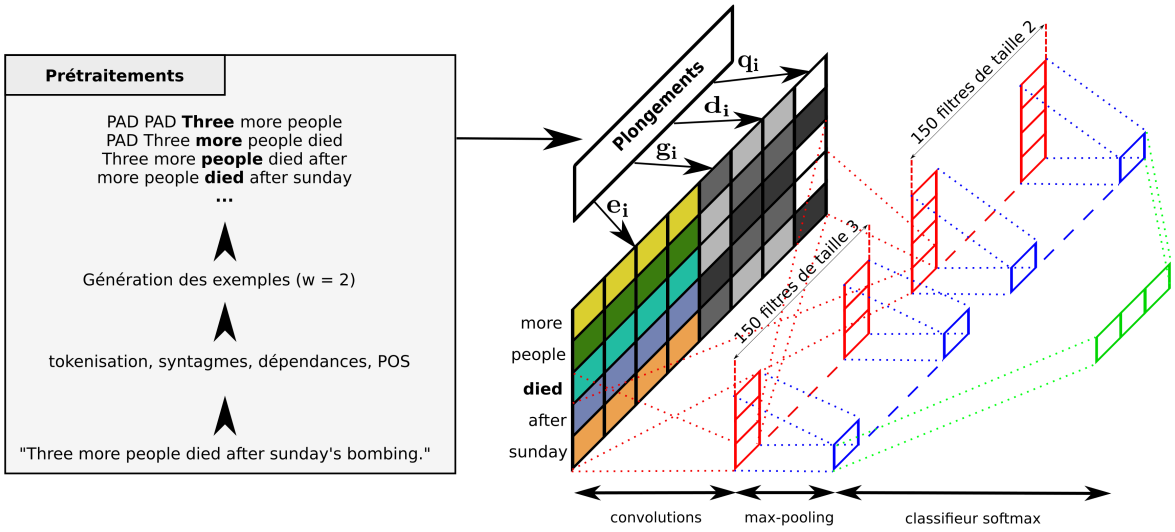

FIGURE 2.1.1 – Illustration d'un modèle convolutif appliqué à la classification d'événements, avec une taille de fenêtre $w = 2$.

Architecture du modèle

Nous présentons en Figure 2.1.1 une illustration simple de notre modèle. Nous considérons successivement chaque mot de chaque phrase en tant que mention candidate. La prédiction d'un candidat peut se décomposer en trois temps, comme nous l'avons montré dans la section 1.5 du précédent chapitre. La première phase a pour rôle de produire une matrice en entrée du modèle convolutif afin de classifier un candidat. Cette mention est représentée par un contexte local de taille fixe centré sur ce mot. Si le contexte local

dépasse les limites de la phrase courante, un token spécial, dit de *padding*, est utilisé pour compléter la séquence. Soit c l'index de la mention candidate et w la taille de la fenêtre. On définit $\mathbf{idx}_c = [c-w, c-w+1, \ldots, c, \ldots, c+w-1, c+w]$ le vecteur des index du contexte local centré sur c. Ce vecteur d'index est transformé en une matrice de réels $\mathbf{X}_c = [\mathbf{x_{c-w}}, \mathbf{x_{c-w+1}}, \ldots, \mathbf{x_c}, \ldots, \mathbf{x_{c+w-1}}, \mathbf{x_{c+w}}]$ en produisant, pour chaque index $i \in \mathbf{idx}_c$, une représentation $\mathbf{x_i} = [\mathbf{e_i}, \mathbf{d_i}, \mathbf{g_i}, \mathbf{q_i}]$ obtenue en combinant les différentes représentations suivantes :

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

Plongements de position $\mathbf{d_i}$ Ce vecteur encode la distance $i - c$ du mot à la position i par rapport au candidat.

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

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

La deuxième phase consiste à extraire un ensemble de traits plus complexes et contextualisés à partir de $\mathbf{X}_c$. À partir de cette matrice, nous appliquons une couche de convolutions constituée de plusieurs filtres de tailles différentes. Cette étape produit un ensemble de vecteurs, chacun correspondant aux différentes applications d'un même filtre le long de la fenêtre de contexte. Une couche de *max-pooling*, c'est-à-dire la sélection de la valeur maximale de chaque filtre, permet d'obtenir une représentation vectorielle dont la dimension est égale au nombre de filtres. Cette représentation du candidat dans son contexte

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

local peut ensuite être utilisée pour réaliser la prédiction. Cette dernière phase consiste en l'application d'une couche dense comportant autant de neurones que de classes, soit le nombre de sous-types d'événements auquel on ajoute la classe NULLE. L'application de la fonction d'activation softmax à ces neurones permet alors d'obtenir la distribution de probabilités des différentes classes d'événements pour le candidat et d'en déduire un type d'événement unique $\hat{y}_c$ en prenant la classe de probabilité maximale.

2.2 Paramètres et ressources

Nous employons les mêmes paramètres pour l'ensemble des réseaux du reste de ce chapitre, sauf lors de l'étude d'un paramètre spécifique. Plus précisément, nous employons 150 filtres pour chaque taille de champ récepteur parmi {2,3,4,5} et utilisons la tangente hyperbolique (tanh) comme non-linéarité pour cette couche. Les plongements de mots sont initialisés à l'aide des plongements à 300 dimensions *word2vec* pré-entraînés sur le corpus Google News [2]. Leurs poids sont modifiés durant l'apprentissage. Les plongements de syntagmes et de position sont initialisés aléatoirement et de taille 50. Les modèles sont entraînés par descente de gradient stochastique (*SGD*) sur des lots de taille 50 et nous utilisons l'optimiseur Adagrad avec un bornage du gradient (*gradient clipping*) doté d'un seuil à 3. La taille de la fenêtre de contexte local est $w = 15$, c'est-à-dire que nous considérons les 15 mots précédant et les 15 mots suivant le mot cible. Tous les F-mesure rapportés dans ce chapitre et les suivants sont des F-mesures (*micro-average*) produits par le programme d'évaluation officiel de la campagne TAC. Le jeu de validation est utilisé pour déterminer le nombre d'époques d'entraînement optimal (*early-stopping*).

Jeux de données

Les expériences réalisées dans ce chapitre l'ont été lors de la campagne d'évaluation TAC Event Nugget 2017, dont le jeu de données de test, $\text{TAC17}_{\text{test}}$ est annoté au format Rich ERE présenté en section 1.3. Plus spécifiquement, 18 des 38 sous-types d'événements

2. `https://code.google.com/archive/p/word2vec/`

du schéma Rich ERE sont annotés. Nous renvoyons les lecteurs à la section 1.3 pour le détail de ces événements. Aucun corpus d'entraînement spécifique n'est fourni mais plusieurs autres jeux de données au format Rich ERE existent :

— **DEFT Rich ERE** (R2 V2 et V2), comportant 288 documents ;

— **TAC15$_{\textbf{[train/test]}}$**, constitué d'un jeu d'entraînement et d'un jeu de test comportant respectivement 158 et 202 documents ;

— **TAC16$_{\textbf{test}}$**, constitué d'un jeu de test de 169 documents annotés de manière similaire à l'édition 2017 (i.e. pour 18 types d'événements).

Dans la section suivante, présentant des expériences préliminaires à la publication du jeu de test TAC 2017, nous étudions notre modèle dans les conditions de la campagne 2016. Notre jeu de test est donc TAC16$_{\text{test}}$ et nous utilisons TAC15$_{\text{test}}$ pour la validation. Nous constituons un jeu d'entraînement hybride, **DEFT/TAC15$_{\textbf{train}}$**, par fusion des jeux de données DEFT Rich ERE (R2 V2 et V2) et TAC15$_{\text{train}}$.

2.3 Étude des hyperparamètres du modèle convolutif

Nous cherchons dans cette section à évaluer les performances de notre modèle convolutif. Comme mentionné précédemment, cette architecture est inspirée du meilleur modèle de l'édition TAC 2016 (Nguyen *et al.*, 2016b). Nous avons fait plusieurs choix de modélisation différant de l'article d'origine. Les principales différences sont :

— nous supprimons le dropout utilisé sur la couche dense et ajoutons un dropout sur la couche d'entrée, avec un taux de dropout $\rho = 0,8$;

— nous initialisons les plongements de mots de notre modèle à l'aide des plongements CBOW entraînés sur Google News à la place des plongements CBOW concaténés ;

— nous utilisons la convolution classique à la place de la convolution non-consécutive proposée spécifiquement par Nguyen et Grishman (2016).

Avant de nous évaluer sur TAC17$_{\text{test}}$, nous présentons ici les expériences ayant mené à nos deux premiers choix de modélisation [3]. Nous nous plaçons donc dans les conditions de

3. Les performances rapportées dans cette section ne sont pas obtenues avec le programme d'évaluation officiel TAC.

la campagne TAC Event Nugget 2016 et utilisons donc $\text{TAC15}_{\text{test}}$ comme jeu de validation sur lequel réaliser l'étude des hyperparamètres. $\text{DEFT/TAC15}_{\text{train}}$ est utilisé pour l'entraînement. Concernant le choix du type de convolution, nous n'avons pas ré-implémenté la convolution non consécutive car la convolution classique est moins intensive en calculs que la convolution non consécutive.

2.3.1 Choix des plongements de mots

Nous considérons en premier lieu le choix du type de plongements à utiliser. Les différents plongements considérés sont :

— **GoogleNews** Plongements CBOW à 300 dimensions entraînés sur GoogleNews fournis avec *word2vec*[4].

— **fastText** Plongements fastText (Bojanowski *et al.*, 2017) à 300 dimensions pré-entraînés sur Wikipédia[5].

— **CBOW** Plongements CBOW à 400 dimensions produits sur le corpus Gigaword (Graff et Cieri, 2003).

— **Skip-Gram structuré** Version modifiée de skip-gram prenant en compte la position des mots du contexte, proposée dans Ling *et al.* (2015). Nous entraînons ces plongements avec 300 dimensions sur le corpus Gigaword.

— **CWindow** Version modifiée de CBOW prenant en compte la position des mots du contexte en concaténant leurs représentations au lieu de les sommer, également proposée par Ling *et al.* (2015). Nous produisons également ces plongements en 300 dimensions sur Gigaword.

Les plongements que nous entraînons nous-mêmes le sont avec les paramètres identifiés par Baroni *et al.* (2014); Levy *et al.* (2015).

Nous présentons en Table 2.3.1 les performances obtenues par notre modèle en fonction des plongements utilisés pour l'initialisation de celui-ci. Ces plongements sont ensuite

4. `https://code.google.com/archive/p/word2vec/`
5. `https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md`

Plongements	F
GoogleNews	**58,8**
Skip-Gram structuré	58,4
Cwindow	55,3
CBOW	55,9
fastText	56,9

TABLE 2.3.1 – Étude de l'influence des plongements pré-entraînés sur les performances du modèle convolutif sur le jeu $TAC15_{test}$. Les modèles sont entraînés sur le corpus $DEFT/TAC15_{train}$.

dropout		
plongements	prédiction	F
Faux	Faux	57,43
Faux	Vrai	56,79
Vrai	Faux	**60.98**
Vrai	Vrai	57,36

TABLE 2.3.2 – Étude de l'influence de différentes configurations de dropout. Les modèles sont évalués sur $TAC15_{test}$ et entraînés sur le corpus $DEFT/TAC15_{train}$.

modifiés durant l'apprentissage. Il apparaît clairement dans la Table 2.3.1 que les plongements GoogleNews offrent les meilleures performances. Ils sont notamment supérieurs aux plongements CWindow. Or, ces plongements, en tant que version concaténée des plongements CBOW, semblent théoriquement comparables à ceux utilisés dans Nguyen *et al.* (2016b) et également entraînés sur GigaWord. Nous n'aboutissons donc pas aux mêmes conclusions que Nguyen *et al.* (2016a) qui mettent en avant l'intérêt de ces plongements. Nous conserverons donc par la suite les plongements GoogleNews.

2.3.2 Dropout

Dans le modèle initial de Nguyen *et al.* (2016b), une couche de dropout est utilisée avant la couche de prédiction. Ce positionnement étant inhabituel et les expériences préliminaires affichant une variance très élevée, nous comparons ce choix au positionnement classique de la couche de dropout après la couche de plongement. Nous considérons donc en Table 2.3.2 quatre cas de figures en fonction de la présence ou absence de dropout sur la couche d'entrée et la couche de prédiction.

Nous constatons que la meilleure configuration obtenue est bien celle de référence,

c'est-à-dire l'inclusion du dropout seulement après les plongements. Il est toutefois possible que la position du dropout dans (Nguyen *et al.*, 2016b) ne soit pertinente que pour une utilisation conjointe avec la convolution non-consécutive que nous n'avons pas réimplémentée.

2.4 Comparaison avec l'état de l'art

Dans cette section, nous nous comparons dans un premier temps aux modèles de la campagne d'évaluation 2016 afin de confirmer la pertinence de nos choix de modélisation. Dans un second temps, nous entraînons de nouveau le modèle et présentons les résultats obtenus par notre modèle dans le cadre de la campagne 2017.

Les jeux de données $\text{TAC16}_{\text{test}}$ et $\text{TAC17}_{\text{test}}$ ne comportent que 18 des 38 sous-types inclus dans la taxonomie Rich ERE et présents dans les autres jeux de données que nous utilisons, notamment pour l'entraînement. Nous entraînons donc nos modèles sur l'ensemble de ces 38 sous-types mais ne conservons que les prédictions pour les 18 sous-types lorsque nous nous évaluons sur TAC 2016 et TAC 2017. Les autres prédictions sont remplacées par la classe NULLE lors de l'évaluation.

2.4.1 TAC 2016

Nous fournissons ici les performances du modèle de la section précédente et entraîné sur $\text{DEFT/TAC15}_{\text{train}}$.

Nous incluons pour comparaison les deux meilleurs modèles sur la tâche de détection d'événements de la campagne TAC 2016 ainsi que le modèle convolutif dont nous nous sommes inspiré. Les modèles présentés sont :

— **CNN** Notre modèle convolutif ;

— **NC-CNN** Le modèle convolutif non consécutif de Nguyen *et al.* (2016b), à l'origine de notre modèle ;

— **UTD** Le système de Lu et Ng (2016) est constitué d'un ensemble de 4 modèles de plus proches voisins opérant sur différents traits lexicaux et syntaxiques ainsi que

des types d'entités de la phrase, détectés automatiquement ;

— **CMU-LTI** Ce modèle, (Liu *et al.*, 2016c), est un CRF s'appuyant sur un ensemble de traits lexicaux et syntaxiques ainsi que sur WordNet.

Les modèles NC-CNN et UTD utilisent, comme nous, les jeux de données DEFT Rich ERE (V2 et R2 V2) en plus des données TAC 2015. Les résultats présentés en Table 2.4.1

Configuration	F
CNN	47,36
NC-CNN	44,37
UTD	46,99
CMU-LTI	44,61

TABLE 2.4.1 – Comparaison des performances de notre modèle sur $\text{TAC16}_{\text{test}}$. Notre modèle est entraîné sur $\text{DEFT/TAC15}_{\text{train}}$.

dépassent les performances du meilleur modèle de la campagne TAC 2016. Il obtient de plus des performances très supérieures au modèle NC-CNN malgré la convolution non-consécutive de ce modèle.

Les modèles NC-CNN et UTD utilisant les mêmes données et le même outil d'analyse morphosyntaxique que nous, la différence de performances avec ceux-ci provient nécessaire de notre modèle. Concernant le modèle CMU-LTI, n'utilisant que les données de la campagne 2015, nous ne pouvons l'affirmer aussi directement mais la différence de performances de $2,75$ points est tout de même suffisamment importante pour le suggérer.

2.4.2 TAC 2017

Les résultats précédents confirmant l'intérêt de nos choix de modélisation, nous conservons cette configuration pour la campagne d'évaluation TAC 2017. Dans ce cadre, nous entraînons d'abord notre modèle sur le jeu d'entraînement constitué de la concaténation de $\text{DEFT/TAC15}_{\text{train}}$ et $\text{TAC15}_{\text{test}}$ et utilisons $\text{TAC16}_{\text{test}}$ pour déterminer le nombre optimal d'époques. Pour le modèle final, nous incluons $\text{TAC16}_{\text{test}}$ dans notre jeu d'entraînement puis entraînons un nouveau modèle sur ce jeu d'entraînement étendu en utilisant le nombre d'époques préalablement déterminé. Nous présentons en Table 2.4.2 les performances de notre modèle, ayant obtenu la troisième place lors de la campagne, ainsi que

les 2 meilleurs modèles de la campagne et le score médian :

1. **Méthode d'ensemble BiLSTM CRF** : Jiang *et al.* (2017) emploient un ensemble de 10 modèles BiLSTM combinés par une stratégie de vote. Mettant en avant le bon rappel des modèles neuronaux au détriment de la précision, ils y adjoignent un classifieur CRF pour améliorer la précision. Pour le BiLSTM, seuls des plongements de mots sont utilisés tandis que le CRF exploite de multiples attributs tels que tokens, lemmes, racines, présence d'entités nommées et étiquettes morphosyntaxiques.
2. **BiLSTM à large marge** : Makarov et Clematide (2017) utilisent un BiLSTM doté d'un objectif à large marge (Gimpel et Smith, 2010). Cet objectif pénalise plus fortement les faux négatifs afin de compenser la rareté des classes positives dans le jeu de données. Un ensemble de 5 réseaux est employé pour la prédiction et des types hybrides sont utilisés.
3. **CNN** : Notre modèle final, entraîné sur DEFT/TAC15$_{\text{train}}$, TAC15$_{\text{test}}$ et TAC16$_{\text{test}}$

Configuration	P	R	F
BiLSTM CRF	56,83	55,57	56,19
BiLSTM à large marge	52,16	48,71	50,37
CNN	54,27	46,59	50,14
Médiane	59,00	40,48	47,94

TABLE 2.4.2 – Résultats sur le jeu de test TAC17$_{\text{test}}$.

Notre modèle obtient des performances très proches du deuxième modèle alors que celui-ci est un modèle d'ensemble de BiLSTM. On observe cependant une importante différence de performances avec le premier modèle, lui aussi constitué d'un ensemble de modèles. Il existe donc une importante marge de progression au niveau des performances atteignables sur ce jeu de données. Nous analysons donc plus en détail les performances de notre modèle et les difficultés de la tâche dans les sections suivantes.

2.5 Étude de la taxonomie des classes d'événements

Nous ne nous sommes, dans les sections précédentes, intéressé qu'à la performance finale de notre modèle sur la tâche de classification des événements en sous-types. Nous

rappelons que la taxonomie Rich ERE est en fait constituée de deux niveaux de granularité : 38 sous-types appartenant à 9 types. De plus, l'identification binaire des déclencheurs (c'est-à-dire la distinction entre les mots déclencheurs et les non-déclencheurs, indépendamment du type d'événement) peut être assimilée à un niveau de granularité plus élevé de cette taxonomie. Cette taxonomie laisse ainsi apparaître deux tâches préliminaires sous-jacentes (identification binaire et classification en type), généralement traitées de manière jointe. Pour la première sous-tâche d'identification binaire, nous observons une forte corrélation entre le classement des modèles sur cette tâche et sur la tâche finale de classification en sous-types dans la présentation des résultats des campagnes (Mitamura *et al.*, 2016, 2017). Si nous pouvons aisément supposer que cette corrélation est en fait causale, le sens de cette causalité n'est pas évident : est-il nécessaire de discriminer génériquement les déclencheurs des non-déclencheurs pour réaliser la classification en sous-types ou, au contraire, est-ce la capacité à distinguer individuellement les différents sous-types qui permet une bonne identification générique ? En d'autres termes, nous souhaitons savoir s'il est possible d'entraîner un modèle capable de produire des attributs génériques discriminant les événements des non-événements.

Bien que nous ne disposions pas de résultats similaires pour la classification en types, nous supposons une corrélation similaire et étendons le précédent questionnement à ce niveau de granularité.

Afin de répondre à notre question, nous appliquons une procédure d'apprentissage incrémental taxonomique (*taxonomic curriculum transfer learning*), telle que proposée par Marino (2016). Dans une tâche de classification où il existe différents niveaux de granularités des classes, l'apprentissage incrémental taxonomique consiste à entraîner un modèle sur les classes grossières (*coarse-grained*) afin d'apprendre des descripteurs génériques. Les poids de ces modèles permettent ensuite d'initialiser un nouveau modèle entraîné sur des classes plus fines (*fine-grained*), comme illustré dans la Figure 2.5.1. Ce processus peut être répété successivement sur des classes de plus en plus fines au sein de la taxonomie. Les auteurs montrent que cette procédure peut être utile pour améliorer les performances lorsque les classes fines sont moins présentes que les classes grossières.

Puisque nous prédisons la classe de chaque token du document et que la large majorité des tokens ne sont pas déclencheurs d'un type existant, on observe que plus de 90% des tokens sont associés à la classe NULLE. Les 10% restants sont répartis entre 38 classes fortement déséquilibrées entre elles et on peut supposer que ce déséquilibre important nuit à la qualité de l'apprentissage. En suivant cette idée, nous cherchons ici à savoir si la prédiction sur des classes fines très déséquilibrées peut tirer profit de descripteurs appris sur des classes grossières plus équilibrés. Si nous observons un gain en utilisant cette méthodologie, nous pourrons conclure qu'il est bien possible de produire des attributs génériques et donc que les tâches amont peuvent gagner à être considérées individuellement. Dans le cas contraire, nous déduirons que la prise en compte des spécificités de chaque classe fine (sous-type) est nécessaire pour l'identification binaire ou la classification en types et qu'il n'est donc pas souhaitable de découpler ces tâches.

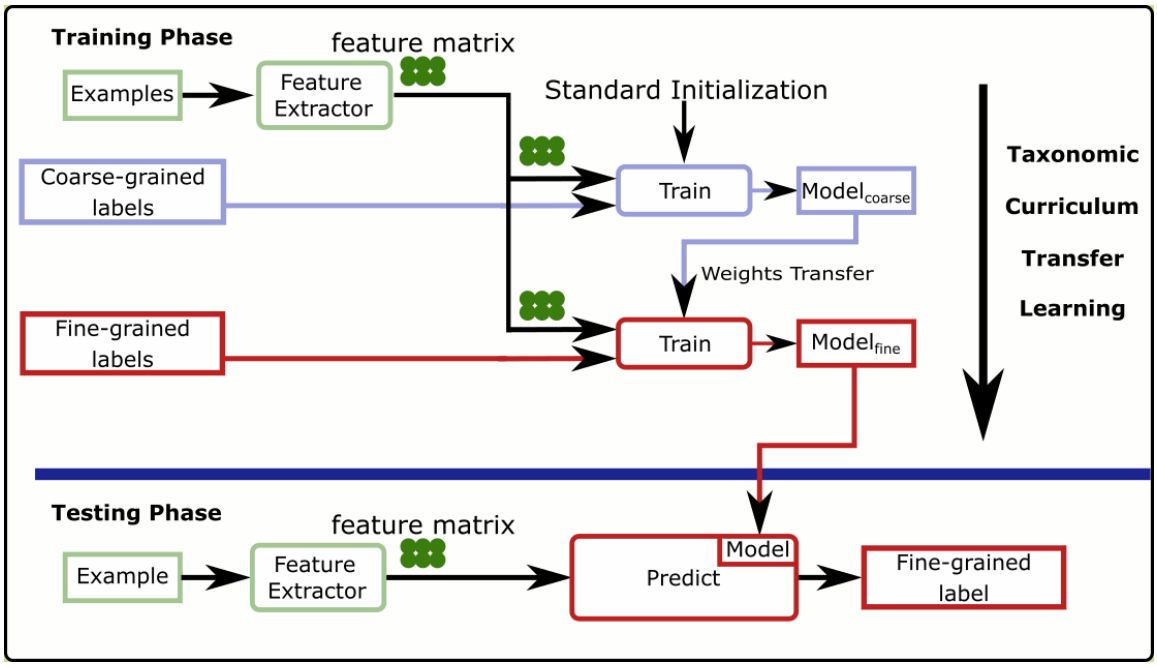

FIGURE 2.5.1 – Illustration de l'apprentissage itératif taxonomique. Dans notre cas, les classes grossières sont soit les classes binaires, soit les 9 TYPES et les classes fines sont les 38 types de la tâche finale.

Jusqu'à la fin de cette section, nous utiliserons "TYPE" et "type" pour faire respectivement référence aux types et sous-types d'événements. Nous avons considéré trois procédures d'entraînement différentes :

CNN$_{type}$ Notre modèle de base, présenté dans la section précédente et qui consiste à

entraîner directement un modèle convolutif sur les 39 classes (38 types et la classe NULLE) de la tâche finale.

$\text{CNN}_{\text{TYPE}\rightarrow\text{type}}$ Un premier modèle, CNN_{TYPE}, entraîné à prédire 10 classes grossières (9 TYPES et la classe NULLE), est ensuite utilisé pour initialiser les poids de $\text{CNN}_{\text{TYPE}\rightarrow\text{type}}$ qui est entraîné sur les 39 classes fines.

$\text{CNN}_{\text{mention}\rightarrow\text{type}}$ On entraîne cette fois un modèle $\text{CNN}_{\text{mention}}$ sur la tâche de classification binaire associée au niveau "mention" puis, comme pour le modèle précédent, on utilise les poids de ce premier modèle comme initialisation d'un nouveau modèle $\text{CNN}_{\text{mention}\rightarrow\text{type}}$ entraîné sur la tâche finale.

2.5.1 Évaluation lors de la campagne TAC 2017

Nous présentons en Table 2.5.1 les résultats de ces trois modèles sur le jeu de test $\text{TAC17}_{\text{test}}$.

Configuration	mention	TYPE	type
CNN_{type}	59,95	57,29	50,14
$\text{CNN}_{\text{TYPE}\rightarrow\text{type}}$	59,57	56,79	49,66
$\text{CNN}_{\text{mention}\rightarrow\text{type}}$	58,32	55,64	49,22
CNN_{TYPE}	59,27	55,47	–
$\text{CNN}_{mention}$	54,09	–	–

TABLE 2.5.1 – Détail des performances de nos modèles pour chaque niveau de granularité sur $\text{TAC17}_{\text{test}}$.

La Table 2.5.1 détaille les performances de nos modèles aux trois niveaux de granularités considérés : la détection binaire de la présence d'un événement (mention), la classification au niveau intermédiaire (TYPE) et la classification fine des événements (type). Nous fournissons également les performances des modèles CNN_{TYPE} et $\text{CNN}_{mention}$, les modèles ayant respectivement servi à initialiser $\text{CNN}_{\text{TYPE}\rightarrow\text{type}}$ et $\text{CNN}_{\text{mention}\rightarrow\text{type}}$.

On constate que l'étape de pré-entraînement affecte négativement les performances du modèle final, et ce, pour l'ensemble des tâches : le modèle entraîné uniquement sur la tâche finale de classification en types est non seulement meilleur sur cette tâche mais obtient également de plus hautes performances sur les autres étapes que les modèles entraînés

Configuration	mention	TYPE	type
CNN_{type}	68.86	66.72	60.98
$\text{CNN}_{\text{TYPE}\rightarrow\text{type}}$	68.41	66.41	60.81
$\text{CNN}_{\text{mention}\rightarrow\text{type}}$	66.63	64.84	60.32
CNN_{TYPE}	68.86	66.93	–
CNN_{detect}	63.67	–	–

TABLE 2.5.2 – Étude des performances des différents schémas d'apprentissage lorsque le jeu de test comporte les mêmes classes que le jeu d'apprentissage. Les modèles sont entraînés sur $\text{DEFT/TAC2015}_{\text{train}}$ et testés sur $\text{TAC15}_{\text{test}}$.

spécifiquement sur celles-ci. Selon la même tendance, CNN_{TYPE} est plus efficace que $\text{CNN}_{mention}$ pour la tâche de détection binaire (mention). Nous en déduisons qu'il n'est pas plus facile de réaliser en amont les tâches de détection binaire et de TYPE d'événement, ni d'exploiter ces tâches pour améliorer les performances de la tâche finale. Il semble donc qu'il y ait autant de différences entre des mentions de différents sous-types d'événements qu'entre mentions d'événements et de la classe NULLE. La même conclusion peut être faite concernant les différences au sein et entre les TYPES.

2.5.2 Évaluation sur TAC 2015

Avant de conclure l'analyse de cette expérience, nous réalisons une expérience similaire sur $\text{TAC15}_{\text{test}}$ afin de déterminer si la différence du nombre de classes entre les données d'apprentissage et de test influence négativement le schéma d'apprentissage taxonomique. Nous présentons donc en Table 2.5.2 les performances par sous tâches sur la base $\text{TAC15}_{\text{test}}$ test des différents schémas d'apprentissage entraînés sur $\text{DEFT/TAC15}_{\text{train}}$, ces jeux de données étant tous deux annotés avec l'ensemble des 38 sous-types d'événements Rich ERE.

Nous en tirons les mêmes conclusions que précédemment, le meilleur modèle sur la tâche finale (type) étant toujours CNN_{type}. On notera toutefois qu'ici le modèle CNN_{TYPE} obtient des performances comparables et même légèrement meilleures sur la tâche TYPE. Ceci peut s'expliquer par la présence du TYPE "Justice" très homogène, donc plus facilement discriminable.

2.6 Limites et Perspectives

Nous avons dans ce chapitre implémenté une architecture convolutive permettant d'obtenir les meilleures performances pour un modèle simple (par opposition aux modèles d'ensembles) sur les éditions 2016 et 2017 de la campagne d'évaluation TAC. Nous avons ensuite mis en avant la difficulté de la tâche en montrant qu'il n'était pas efficace de réaliser des classifications préalables plus grossières et que la résolution de la tâche nécessitait de distinguer finement chaque sous-type d'événements des autres et de la classe NULLE. Par ailleurs, les performances du modèle d'ensemble intra-phrastique de Jiang *et al.* (2017) présentent des performances bien supérieures. Nous en concluons que notre modèle n'est pas en mesure de tirer pleinement parti des informations exploitables au niveau phrastique.

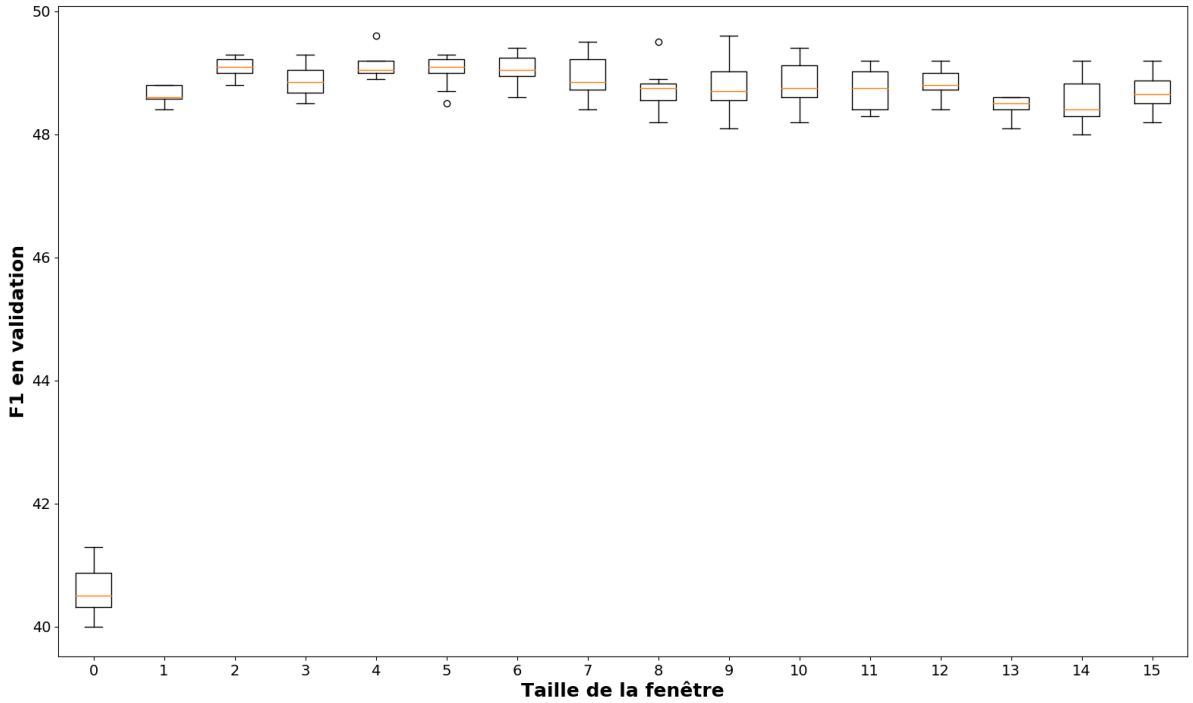

FIGURE 2.6.1 – Performances du modèle convolutif sur $\text{TAC16}_{\text{test}}$ en fonction de la taille de la fenêtre de contexte.

Afin d'observer la capacité du modèle à exploiter des dépendances longues au sein de la phrase, nous présentons à la Figure 2.6.1 les performances du modèle en fonction de la taille de ce contexte. Nous faisons pour cela varier le paramètre w représentant la taille de la fenêtre de contexte considérée autour du mot cible. Nous étudions la plage

$[0, 15]$, 15 étant la taille utilisée par les modèles présentés jusqu'ici dans ce chapitre ainsi que par (Nguyen *et al.*, 2016b) et 0 consistant à ne considérer que le mot cible. Nous constatons que les performances du CNN saturent dès $w = 2$ (c'est-à-dire seulement deux mots à gauche et deux mots à droite du mot cible). Or, il ne fait aucun doute que des informations pertinentes se trouvent au-delà de cette distance. Ces résultats indiquent que le modèle convolutif ne parvient pas véritablement à exploiter des dépendances distantes au sein du contexte intra-phrastique. Et comme nous l'avons vu, ce modèle local obtient des performances similaires à celles de l'ensemble de BiLSTM de Makarov et Clematide (2017). Il semble donc que cette architecture récurrente, bien que théoriquement mieux à même d'exploiter des dépendances longues, souffre également de cette difficulté à exploiter les dépendances longues au niveau intra-phrastique. Ces observations constituent la motivation première des contributions présentées dans les prochains chapitres : malgré leurs performances satisfaisantes, les architectures neuronales classiques ne sont pas en mesure d'exploiter pleinement le contexte intra-phrastique distant. Nous proposons des réponses à ces limites à travers la prise en compte du contexte inter-phrastique dans les chapitres 3, 4 et 5 ainsi que par une meilleure prise en compte du contexte intra-phrastique dans le chapitre 4.

Chapitre 3

Génération par amorçage d'une représentation de documents spécifique à la tâche

Sommaire

3.1 Problématique

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

Ceci illustre l'importance de la prise en compte du contexte pour résoudre les différentes ambiguïtés possibles entre différents types. Cependant, comme nous l'avons montré dans le chapitre précédent, notre modèle convolutif n'est pas en mesure d'exploiter un contexte intra-phrastique long. Puisque nous obtenons des performances comparables à des modèles BiLSTM, réputés capable d'exploiter des dépendances longues au sein de séquences, cette limitation n'est donc pas circonscrite aux architectures convolutives. De ce fait, lorsque le contexte intra-phrastique proche n'est pas assez informatif pour permettre au modèle de discriminer convenablement un type d'événements d'un autre ou de l'absence d'événement, des ambiguïtés peuvent subsister. Nous distinguons ainsi deux types d'ambiguïtés en fonction de la portée du contexte nécessaire à leur résolution :

— « Le rappeur lyonnais **déclara** "Les instruments c'était mieux à vent" aux micros

de France Inter. »

— « Les **départs** se multiplient chez l'opérateur téléphonique. [...] Plus de 200 démissions ont ainsi été reçues le mois dernier. »

Dans le premier exemple, l'ambiguïté fréquente entre les types d'événements *Contact-contact* et *Contact-broadcast* peut être résolue en identifiant que la communication s'adresse à un média et se fait donc à sens unique. Mais la distance entre *déclara* et *France Inter* peut rendre cette information inaccessible bien qu'elle soit locale à la phrase. Dans le second exemple, le contexte local de la phrase n'indique pas clairement que *départ* fait référence à un événement de type *End_Position*. Mais la thématique des licenciements est clairement identifiable plus loin dans le document. Il s'agit ici d'un problème de désambiguïsation inter-phrastique.

Nous avons identifié dans le chapitre précédent l'incapacité des modèles neuronaux actuels à exploiter un contexte intra-phrastique long. Cette limite conduirait logiquement dans un premier temps à étudier des manières de mieux prendre en compte ce contexte intra-phrastique. Toutefois, si l'intérêt des modèles intra-phrastiques est d'exploiter finement une plus grande richesse d'informations linguistiques, et notamment de nature syntaxique, il n'est pas évident d'affirmer que, même en parvenant à exploiter l'information distante, cette finesse ne s'atténue pas avec la longueur des dépendances. De plus, cette approche ne résoudrait que le premier type d'ambiguïtés alors qu'il existe un nombre non négligeable de cas nécessitant de prendre également en compte le contexte inter-phrastique. Ce constat justifie l'intérêt d'une optique inter-phrastique sachant que cette optique peut aussi fournir des informations utiles pour les cas d'ambiguïté intra-phrastique et permet d'avoir une approche unifiée de ces deux types d'ambiguïtés.

Pour ce faire, nous nous intéressons donc à la prise en compte d'un contexte plus global à travers la production et l'intégration d'une représentation vectorielle de ce contexte. Duan *et al.* (2017) mettent en avant l'intérêt d'une telle exploitation afin de prendre en compte la cohérence interne des documents sur le plan thématique : un document traitant d'un conflit armé présentera plus d'événements de type *Die* ou *Attack* que de naissances. Ils proposent de fournir en entrée d'un BiLSTM une représentation distribuée du docu-

ment apprise de manière non supervisée (Le et Mikolov, 2014). Ce contexte global n'est donc pas spécialisé pour la tâche cible. Notre approche vise au contraire à apprendre une représentation des documents en lien avec la tâche cible et pouvant résoudre les deux types d'ambiguïtés identifiés précédemment. Nous utilisons pour ce faire une méthode d'amorçage en définissant un premier modèle à un niveau local (niveau des tokens) et en l'appliquant à l'ensemble du document. Les prédictions locales ainsi réalisées sont agrégées pour obtenir un vecteur de contexte pour chaque document. Ces vecteurs sont alors intégrés à un nouveau modèle exploitant ce contexte. Dans la suite de ce chapitre, nous présentons dans un premier temps nos modèles local et global à la section 3.2. La section 3.3.2 compare les différents types de représentations du contexte global utilisables. Enfin, nous évaluons à la section 3.3.3 différentes modalités d'intégration de cette représentation globale au modèle local et les confrontons à plusieurs baselines sur les données de la campagne d'évaluation TAC Event Nugget 2017. Ces expériences montrent que ce nouveau modèle obtient des gains significatifs par rapport au modèle local *baseline* et s'avère compétitif par rapport à d'autres approches neuronales. En dernier lieu, nous étudions plus en détail l'apport de la représentation globale à la section 3.4.

3.2 Description de l'approche

Notre approche consiste à prédire l'ensemble des déclencheurs d'un document à l'aide d'un modèle local avant d'agréger cette information sous la forme d'un vecteur intégré dans un second modèle. Ces deux modèles sont des modèles convolutifs de même architecture que dans le chapitre précédent. Nous rappelons donc dans un premier temps cette architecture puis décrivons la procédure d'extraction et d'intégration de notre représentation, que nous illustrons en Figure 3.2.1, dans la suite de cette section.

3.2.1 Architecture locale

Dans ce chapitre, nous nous replaçons comme précédemment dans les conditions de la campagne d'évaluation TAC 2017. Nous rappelons que, puisque les mentions d'évé-

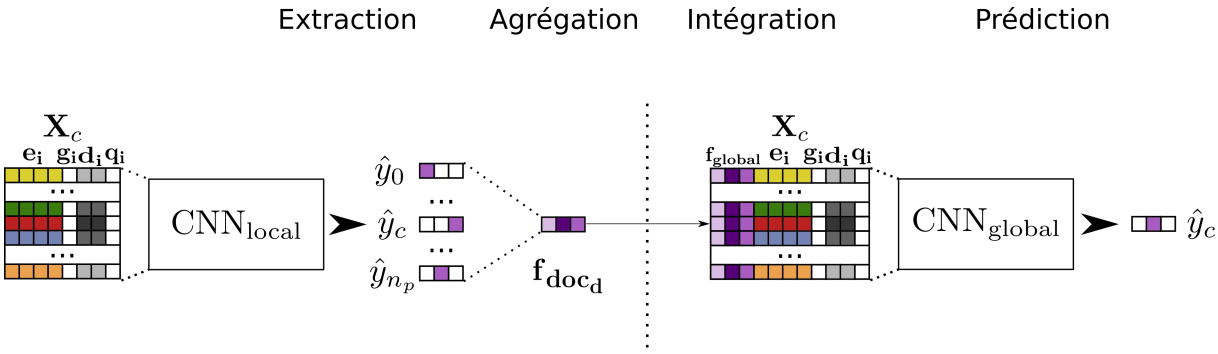

FIGURE 3.2.1 – Principe d'intégration du contexte par amorçage. Un premier modèle $\text{CNN}_{\text{local}}$ est entraîné pour associer des étiquettes d'événements à chaque mot d'un document. Ces étiquettes sont agrégées au niveau d'un contexte et ajoutées en entrée d'un nouveau modèle $\text{CNN}_{\text{global}}$. $\hat{y}_c$ est la prédiction du modèle pour la mention courante, $\hat{y}_0$, la prédiction pour la première mention trouvée du document et $\hat{y}_{n_p}$, la dernière mention trouvée du document.

nements sont en grande majorité des mots simples (Reimers et Gurevych, 2017), nous choisissons d'aborder le problème non pas comme une tâche d'annotation de séquences mais comme une tâche de classification multi-classe de mots. Ce choix est d'un impact négatif négligeable mais simplifie la modélisation et permet l'introduction d'un vecteur de positions contribuant grandement aux performances (Nguyen *et al.*, 2016b). Enfin, dans la continuité du précédent modèle, nous opérons à l'échelle intra-phrastique. La tâche est alors envisagée comme un problème de classification multi-classe.

3.2.2 Modèle local de détection d'événements

Au niveau local, notre modèle convolutif de détection d'événements est celui présenté dans le chapitre précédent (2.1). Nous ne le décrivons donc que succinctement en introduisant quelques notations complémentaires et renvoyons le lecteur à la section correspondante pour la présentation détaillée du modèle. Pour un candidat d'index c et une fenêtre de taille w. Nous définissons $\mathbf{idx}_c = [c-w, c-w+1, \ldots, c, \ldots, c+w-1, c+w]$ le vecteur des index du contexte local centré sur c. Pour chaque index $i \in \mathbf{idx}_c$, nous produisons une représentation :

$$\mathbf{x_i} = [\mathbf{e_i}, \mathbf{d_i}, \mathbf{g_i}, \mathbf{q_i}] \tag{3.1}$$

par concaténation des plongements de mots (e_i), positions (d_i), dépendances (g_i) et syntagmes (q_i). Nous obtenons ainsi la matrice d'entrée :

$$\mathbf{X}_c = [\mathbf{x_{c-w}}, \mathbf{x_{c-w+1}}, \ldots, \mathbf{x_c}, \ldots, \mathbf{x_{c+w-1}}, \mathbf{x_{c+w}}]$$

L'application successive des convolutions permet d'obtenir une représentation $\mathbf{f_{pooling}}$. Nous appelons la représentation en entrée de la couche de prédiction $\mathbf{f_{softmax}}$. Dans le modèle local, nous fournissons directement la représentation de la couche de convolution à la couche de prédiction :

$$\mathbf{f_{softmax}} = [\mathbf{f_{pooling}}] \tag{3.2}$$

Le label prédit, $\hat{y}_c$, est alors la classe de probabilité maximale.

3.2.3 Intégration de contexte dans un modèle global

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

Nous utilisons la notation suivante pour désigner les différentes configurations possibles de contexte global d'une phrase : $\mathbf{f_{global}} = \mathbf{f_{[doc/large/phrase]}}$. Le contexte global $\mathbf{f_{global}}$ peut être intégré au niveau de la matrice d'entrée $\mathbf{X}_c$ en redéfinissant 3.1 ou avant la couche

entièrement connectée en redéfinissant 3.2 :

$$\textbf{Plongement} \quad \mathbf{x_i} = [\mathbf{e_i}, \mathbf{d_i}, \mathbf{g_i}, \mathbf{q_i}, \mathbf{f_{global}}] \tag{3.3}$$

$$\textbf{Softmax} \quad \mathbf{f_{softmax}} = [\mathbf{f_{pooling}}, \mathbf{f_{global}}] \tag{3.4}$$

Nous distinguerons ainsi six modèles en fonction des niveaux d'agrégation et d'intégration avec la notation $\text{CNN}_{\text{[doc/large/phrase]-[plongement/softmax]}}$.

3.3 Expériences

3.3.1 Paramètres et ressources

Nous utilisons dans nos expériences les *plongements* à 300 dimensions pré-entraînés sur Google News avec *word2vec* en les modifiant durant l'entraînement. Les vecteurs de positions et de syntagmes sont de taille 50. La probabilité de *dropout* est fixée à 0,8. Pour chacune des tailles de champ récepteur (2,3,4,5), 150 filtres sont utilisés, pour un total de 600. Ces filtres sont dotés d'une tangente hyperbolique comme non-linéarité. Le modèle est entraîné par descente de gradient stochastique (SGD) avec l'optimiseur Adagrad et un *clipping* du gradient fixé à 3. La taille des mini-lots est fixée à 50. Le nombre d'époques d'apprentissage est contrôlé par *early stopping* sur le jeu de validation. Les résultats présentés sont des moyennes sur 10 exécutions. Nous nous évaluons dans les mêmes conditions que les expériences du chapitre précédent sur l'édition 2017 : notre corpus d'entraînement est constitué de l'union des jeux de données DEFT Rich ERE R2 V2, DEFT Rich ERE V2 et TAC 2015. Notre corpus de validation est le jeu de données issu de la campagne TAC 2016 ($\text{TAC16}_{\text{test}}$) et nous nous testons sur les données de la campagne TAC 2017 Event Nugget ($\text{TAC17}_{\text{test}}$). Il existe au sein de ces jeux de données quelques rares cas de mentions annotées avec plusieurs types d'événements distincts. Parmi ces cas de figures, la grande majorité appartient à l'une des trois combinaisons suivantes : (*Attack/Die*, *Transfer-Money/Transfer-Ownership*, *Attack/Injure*). Pour traiter ce problème, nous introduisons trois types hybrides lors de l'apprentissage pour ces trois types d'événe-

ments, ce qui permet de conserver une classification simple (mono-étiquette). Nos jeux de données de validation et de test se focalisent sur les types d'événements les plus difficiles de Rich ERE, à l'instar de $\text{TAC17}_{\text{test}}$, et restreignent la tâche à 19 des 38 types. Nous entraînons toutefois notre modèle sur l'ensemble des 42 classes (classes hybrides et classe NULLE incluse) mais nous ignorons les types non présents en test lors de la prédiction. De même, le vecteur global n'agrège que les prédictions des types présents sur le jeu de test. Enfin, différentes normalisations du vecteur de contexte global ont été comparées expérimentalement. Les résultats présentés ici reposent sur la meilleure normalisation obtenue en validation pour chaque configuration : les vecteurs $\mathbf{f}_{\mathbf{[large/phrase]}}$ ne sont pas normalisés alors que le vecteur $\mathbf{f}_{\mathbf{doc}}$ est centré-réduit avant d'être fourni au modèle.

3.3.2 Influence de la taille du contexte global

Comme nous l'avons vu précédemment, l'agrégation du contexte peut se faire à plusieurs niveaux. L'agrégation au niveau de la phrase courante peut résoudre les ambiguïtés intra-phrastiques alors qu'une agrégation plus large peut être utile pour la désambiguïsation inter-phrastique. Afin de déterminer le niveau d'agrégation le plus utile, nous comparons en Table 3.3.1 l'apport de l'intégration du contexte $\mathbf{f}_{\mathbf{global}}$ à la matrice d'entrée $\mathbf{X}_c$ en fonction du niveau d'agrégation : *phrase*, *large* et *doc*. Nous constatons tout d'abord que les trois niveaux d'agrégation améliorent les performances par rapport à notre baseline $\text{CNN}_{\text{local}}$. De plus, les performances augmentent avec la taille du contexte, l'agrégation au niveau du document fournissant les meilleures performances ainsi que le seul gain significatif. Ici, et pour les prochains chapitres, nous considérons comme significatif une p-valeur $p < 0,05$ obtenue à l'aide d'un test de Student non-apparié.

Deux hypothèses peuvent être émises concernant ce résultat : soit la prise en compte d'un contexte inter-phrastique plus large permet de lisser les erreurs du modèle local, soit ce contexte plus large est intrinsèquement plus pertinent. L'étude de ces hypothèses est présentée en section 3.4.1.

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

TABLE 3.3.1 – Performances sur la base de validation $\text{TAC16}_{\text{test}}$ en fonction du niveau d'agrégation. Résultats moyennés sur 10 entraînements pour chaque configuration. Seul le modèle $\text{CNN}_{\text{doc-plongement}}$ est significativement meilleur que $\text{CNN}_{\text{local}}$ ($p < 0,01$).

3.3.3 Comparaison avec l'état de l'art

Afin de valider l'apport de notre méthode, en plus de la comparaison à notre modèle initial $\text{CNN}_{\text{local}}$, nous nous comparons aux 3 modèles ayant obtenu les meilleurs résultats lors de la campagne d'évaluation TAC 2017. Ces modèles ayant déjà été décrits dans le chapitre 2, nous le faisons ici de manière concise et renvoyons le lecteur à ce chapitre pour plus de détails.

— **Méthode d'ensemble BiLSTM CRF** Jiang *et al.* (2017) utilisent un ensemble de 10 modèles BiLSTM combinés par une stratégie de vote à un modèle CRF.

— **BiLSTM à large marge** Makarov et Clematide (2017) utilisent un ensemble de 5 BiLSTMs dotés d'un objectif à large marge pénalisant plus fortement les faux négatifs.

— **Modèle convolutif** ce modèle est le CNN présenté dans le chapitre 2. La principale différence avec le modèle convolutif de ce chapitre ($\text{CNN}_{\text{local}}$) est l'absence de types hybrides pour gérer les cooccurrences d'événements.

— $\textbf{CNN}_{\textbf{doc2vec}}$ à l'instar de Duan *et al.* (2017), nous intégrons un vecteur de document au niveau des plongements de notre modèle. Ce vecteur de taille 100 est produit par le modèle PV-DM (Le et Mikolov, 2014). À la différence de notre représentation globale, celle-ci n'est pas spécifique à la tâche. Nous avons optimisé les mêmes hyperparamètres d'intégration que pour notre représentation, à savoir le choix de la normalisation et du niveau d'intégration. La meilleure configuration présentée ici intègre des vecteurs centrés-réduits au niveau du softmax.

La Table 3.3.2 présente la comparaison de ces méthodes sur la base de test $\text{TAC17}_{\text{test}}$.

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

TABLE 3.3.2 – Performances sur $TAC17_{test}$. "†" désigne des modèles d'ensemble. ‡ indique dans la seconde partie de la table les modèles significativement meilleurs que le modèle CNN_{local} ($p < 0,01$ pour un t-test bilatéral sur les moyennes).

Il est difficile de comparer notre contribution à Jiang *et al.* (2017) et Makarov et Clematide (2017) car ce sont des méthodes d'ensemble alors que nous présentons les performances pour un modèle simple. De plus, leurs performances moyennes sur plusieurs initialisations ne sont pas disponibles alors que les variations sont souvent non négligeables (Reimers et Gurevych, 2017).

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

Un autre élément rendant les comparaisons difficiles est la présence de blocs de citations dans les documents provenant de forums de discussions. Ces citations ne sont pas annotées en événements, même lorsqu'elles reprennent des phrases précédentes contenant effectivement des événements. Ces phrases constituent donc des duplicatas de phrases contenant possiblement des événements. Durant l'apprentissage, le modèle reçoit alors des annotations contradictoires pour deux exemples pourtant identiques. Durant le test, ignorer ces phrases peut ainsi significativement augmenter les performances. Les résultats présentés ici ne tiennent pas compte de cet aspect, ce qui minore très certainement les

performances de notre modèle. Nous avons eu par ailleurs confirmation [1] que les performances rapportées par Makarov et Clematide (2017) négligeaient ce phénomène et que sa prise en compte dans leur modèle entraînait également un gain significatif.

Leur approche peut donc être comparée à la nôtre de ce point de vue [2], comparaison montrant que notre *baseline* $\text{CNN}_{\text{local}}$ est légèrement supérieure en moyenne aux performances du BiLSTM à large marge de Makarov et Clematide (2017). Malgré la tendance actuelle à privilégier les architectures récurrentes, les modèles utilisant la convolution restent donc compétitifs. Comme nous l'avons dit dans le chapitre précédent, nous supposons que les RNNs utilisés ne sont pas non plus en mesure d'exploiter réellement l'intégralité du contexte à disposition, l'influence du contexte proche étant prédominante dans la majorité des cas. Enfin, pour achever la comparaison avec les modèles de la campagne TAC, la Table 3.3.2 montre que l'introduction de types hybrides dans notre modèle local ($\text{CNN}_{\text{local}}$) permet d'obtenir de meilleurs résultats que notre précédent modèle convolutif (CNN) présenté dans le chapitre précédent.

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

3.4 Discussions

3.4.1 Analyse du choix du contexte

En premier lieu, il faut souligner que du point de vue de la taille du contexte à prendre en compte, les résultats de la section 3.3.2 montrent assez clairement l'intérêt

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

de se situer à l'échelle du document plutôt qu'à une granularité de contexte plus fine. Une interprétation possible de ce constat est que l'intégration d'un contexte global au modèle est intrinsèquement plus adaptée à la résolution des ambiguïtés inter-phrastiques qu'intra-phrastiques. Sur un autre plan, nous notons également que le contexte global est produit à partir des prédictions d'un premier modèle imparfait. Il est donc bruité. Agréger les prédictions sur un contexte plus large permettrait ainsi de compenser ce bruit plus efficacement. Afin de distinguer ces deux phénomènes, nous comparons dans la Table 3.4.1 les résultats de l'intégration au niveau de la phrase et du document en utilisant cette fois les annotations réelles à la place des prédictions du $\text{CNN}_{\text{local}}$. Nous constatons que cette fois encore, les meilleures performances sont obtenues en agrégeant l'information à l'échelle du document. L'écart avec $\text{CNN}_{\text{phrase-plongement}}$ est même encore plus élevé. Il semble donc que le niveau d'agrégation ne dépend pas des performances du modèle local et que l'agrégation à l'échelle du document soit intrinsèquement meilleure, peut-être en raison d'une possible prévalence des ambiguïtés inter-phrastiques.

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

Concernant le niveau d'intégration du contexte global, les résultats de la Table 3.3.2 montrent que les gains de $\text{CNN}_{\text{doc-plong\_soft}}$ et de $\text{CNN}_{\text{doc-plongement}}$ sont obtenus en privilégiant la précision au détriment du rappel. Au contraire, la configuration la plus favorable, $\text{CNN}_{\text{doc-softmax}}$, permet une amélioration de la précision, certes plus faible, mais ne dégradant pas le rappel. Ce modèle étant le plus favorable, nous nous concentrons sur celui-ci

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

TABLE 3.4.1 – Performances sur la base de validation $TAC16_{test}$ en fonction du niveau d'agrégation avec le contexte parfait. Résultats moyennés sur 10 entraînements pour chaque configuration.

dans le reste de cette étude.

3.4.2 Analyse d'erreur du meilleur modèle

TABLE 3.4.2 – Comparaison détaillée des performances par classe entre CNN_{local} et $CNN_{doc\text{-}softmax}$. Pour une meilleure visibilité, nous rapportons les mesures sans les décimales. Les performances en **P**récision, **R**appel et **F**-mesure sont en gras lorsque le modèle global est meilleur que le modèle local, le nom de la classe l'est seulement quand la F-mesure est meilleure.

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

La Figure 3.4.1 présente un comparatif des performances du modèle local et du modèle $CNN_{doc\text{-}softmax}$ en détaillant les résultats par type d'événements. Pour plus de précision, la Table 3.4.2 donne en outre les valeurs correspondantes. Une première observation est

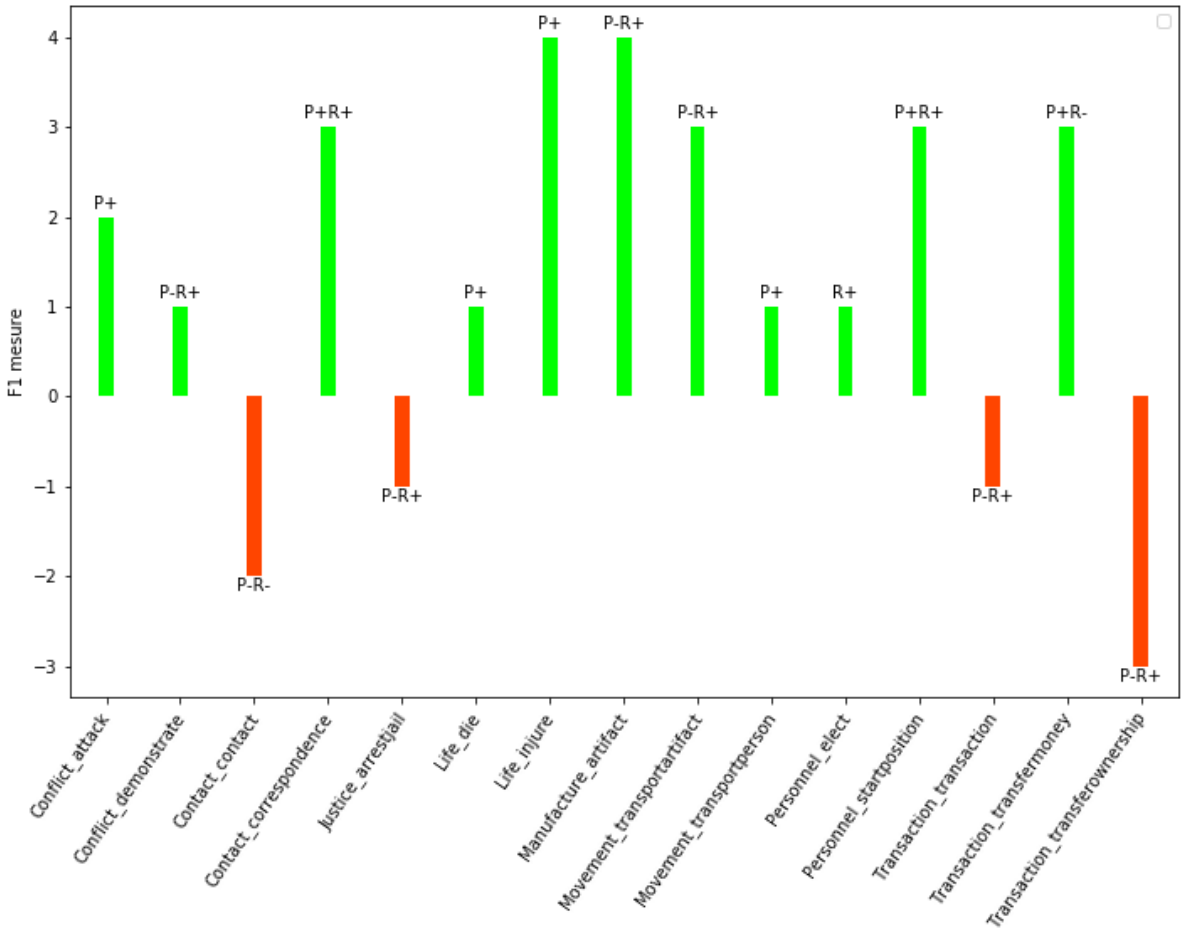

FIGURE 3.4.1 – Variation des F-mesures par classe entre $\text{CNN}_{\text{local}}$ et $\text{CNN}_{\text{doc-softmax}}$ sur $\text{TAC15}_{\text{test}}$. Seules les classes présentant une variation de performance sont présentées. Les variations en **P**récision et en **R**appel sont précisées au-dessus de la barre, "-" indiquant une réduction et "+" une augmentation par rapport au modèle local.

l'existence d'un écart de performance important entre les différentes classes, aussi bien pour le modèle local que global. Puisque notre modèle global s'appuie sur les prédictions du modèle local, nous pourrions craindre que la représentation globale dégrade les performances sur les classes faibles. Nous constatons que ce n'est pas le cas : sur les 5 classes ayant les performances initiales les plus basses, 2 ne sont pas affectées et les performances augmentent pour une classe. Les deux classes restantes, Contact-Contact et Transaction-Transaction, sont les sous-types utilisés respectivement lors de l'annotation en cas d'ambiguïté avec d'autres sous-types des types Contact et Transaction. Dans les deux cas, cette baisse s'accompagne d'une amélioration d'une autre classe du même type d'événement (respectivement Contact-Correspondance et Transaction-Transfer-money), ce qui indique un probable transfert entre ces sous-types.

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

— « Do n't get me wrong , I 'm **[**glad he **won** , I **]** voted for him . »

Contexte global : (elect : 14, correspondence : 5, contact : 3, transport_person : 3, broadcast : 2)

Faux négatif : NULLE (au lieu de elect)

— « 100,000 MtGox **[**Bitcoins were **lost** through theft**]** (about $ 500 million or 7 % of the outstanding Bitcoins). »

Contexte global : (transfer_ownership : 11, transfer_money : 7, contact_contact : 2, transport_person : 2, manufacture_artifact : 2, die : 1, transaction : 1, arrest_jail : 1)

Faux positif : die

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

Conclusion et perspectives

Nous avons dans le chapitre précédent mis en avant l'incapacité du modèle convolutif à exploiter un contexte intra-phrastique distant. Afin de répondre à cette limitation, nous avons proposé une méthode permettant la prise en compte du contexte inter-phrastique à l'échelle du document. Par contraste avec une représentation existante et non spécifique, nous avons émis l'hypothèse qu'une représentation axée sur la tâche d'extraction d'événements est plus à même d'améliorer les performances du modèle. Notre méthode, fondée sur l'agrégation des prédictions d'un premier modèle sous forme d'un vecteur de contexte puis sur l'intégration de ce vecteur à un second modèle, fournit un gain significatif par rapport à notre modèle de base et par rapport à la représentation générique à laquelle nous nous comparons. Nous avons ainsi confirmé notre hypothèse et, par la même occasion, obtenu les meilleures performances d'un modèle simple sur cette tâche (par opposition au modèle d'ensemble de Jiang *et al.* (2017)).

Nous supposons toutefois que cette représentation, qui prend la forme d'un histogramme de la distribution des événements telle qu'estimée par le modèle local, est limitée dans son expressivité. Une première piste d'amélioration de cette contribution serait d'agréger plutôt le vecteur produit par l'étape de pooling, $\mathbf{f_{pooling}}$. Des expériences préliminaires dans ce sens n'ont cependant pas permis d'obtenir d'améliorations par rapport à cette représentation. D'autre part, si l'utilisation d'une représentation de contexte issue d'un modèle entraîné sur la tâche de détection d'événements constitue bien une représentation spécifique, on peut toutefois regretter que cette représentation ne soit pas apprise conjointement avec la prédiction des événements. Nous proposons donc dans le chapitre suivant une autre représentation plus satisfaisante à ces deux niveaux.

Chapitre 4

Extraction dynamique de représentations du contexte par coréférence

Sommaire

L'objectif central de ces travaux de thèse est, pour rappel, la meilleure exploitation du contexte pour améliorer les performances des modèles de détection d'événements. Nous distinguons deux types de contextes : le contexte intra-phrastique, c'est-à-dire l'ensemble des informations contenues au sein de la phrase cible et le contexte inter-phrastique, ou global, qui désigne au contraire l'ensemble des informations contenues dans le reste du document.

Nous avons dans le précédent chapitre proposé une première contribution permettant d'obtenir un gain significatif via l'introduction d'un vecteur de distribution des événements. Toutefois, le potentiel de cette méthode est limité par essence. L'approche par amorçage à deux passes s'appuie en effet sur une représentation peu expressive et non spécifique à la phrase prédite, la représentation étant la même pour l'ensemble du document. De plus l'application de cette méthode nécessite d'appliquer successivement le modèle local puis global au document. Nous proposons dans ce chapitre une nouvelle méthode de prise en compte du contexte global exploitant cette fois-ci une représentation dense et spécifique ne nécessitant qu'une seule passe sur le document à prédire. Pour ce faire, nous introduisons une nouvelle représentation du contexte, utilisant un BiLSTM appliqué aux phrases du document en lien de coréférence avec la phrase à prédire.

Par ailleurs, nous nous intéressons ici également à la meilleure prise en compte du contexte intra-phrastique. De récentes approches montrent l'intérêt des réseaux de neurones de graphes pour la détection d'événements. Ces modèles, à l'inverse des approches classiques (CNN, RNN), ne s'appuient pas sur la proximité de surface (mots précédents et suivants) mais sur les dépendances syntaxiques pour définir le voisinage d'un mot. Ceci permet de réduire la distance entre le déclencheur et les mots importants tels que les entités nommées.

La première section de ce chapitre considère le contexte intra-phrastique en présentant l'architecture d'un modèle de convolution de graphe issu de l'état de l'art. Nous présentons ensuite dans une deuxième partie une proposition de représentation du contexte inter-phrastique consistant en l'application d'un BiLSTM dit de contexte aux phrases en lien avec la phrase cible afin de produire une représentation intégrée au modèle local, ces deux modèles étant entraînés de manière jointe. Enfin, nous testons sur les éditions 2015 et 2017 de la campagne d'évaluation TAC notre réimplémentation du modèle local et le modèle utilisant notre représentation et obtenons avec celui-ci les meilleures performances pour un modèle simple sur ces deux jeux de données.

4.1 Prise en compte du contexte intra-phrastique distant

L'exploitation des informations contenues dans le contexte intra-phrastique joue un rôle capital dans la résolution de la tâche de détection 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. La capacité du modèle convolutif, à la base de notre première contribution, à exploiter un contexte intra-phrastique long étant limitée, nous considérons dans cette section le modèle de convolution de graphe, plus à même d'accomplir cette tâche. Dans un premier temps, nous analysons plus précisément les limites du modèle convolutif et en quoi le modèle de graphe constitue une alternative désirable. Nous détaillons ensuite le fonctionnement de ce modèle local.

4.1.1 Problématique

Bien que les modèles convolutifs permettent d'obtenir des performances satisfaisantes en détection d'événements comme nous l'avons montré précédemment, leur fonctionnement permet seulement l'identification de cooccurrences locales au sein de séquences (ici, des n-grammes au sein de la forme de surface de la phrase). La complexité des dépendances qu'ils peuvent exploiter est donc limitée à un contexte local proche.

Nous faisons ici l'hypothèse que les entités jouent un rôle clé dans la résolution d'ambiguïtés locales et que la capacité à tenir compte des dépendances longues doit avant tout s'entendre comme la capacité à modéliser les interactions entre le déclencheur événementiel et les mentions d'entités de la phrase. Ainsi, dans l'exemple ci-dessous, l'ambiguïté du mot "*fired*" (renforcée par la présence de "*suddenly*" laissant penser à un événement violent) peut être résolue en identifiant qu'elle fait référence à une responsabilité administrative, "*Attorney*". Cependant, la distance entre ces deux mots rend l'apprentissage

d'un filtre modélisant un tel bigramme impossible.

> *The U.S Attorney who was probing a bribery case was suddenly* ***fired*** *by George W. Bush*

La solution à cette limitation peut être formulée de deux manières différentes : permettre la prise en compte de dépendances plus longues au sein de la séquence ou opérer sur une représentation dans laquelle l'information contextuelle distante dans la forme de surface devient plus proche.

Les modèles récurrents sont théoriquement capables d'exploiter des dépendances longues au sein de séquences et semblent donc de bons candidats pour implémenter la première forme de solution. Cependant, comme nous l'avons montré dans le chapitre 1, la comparaison des performances de modèles récurrents (Duan *et al.*, 2017; Feng *et al.*, 2016) avec des modèles convolutifs (Chen *et al.*, 2017; Nguyen et Grishman, 2016) ne laisse pas apparaître de grandes différences.

Nous nous intéressons donc à la seconde solution, en utilisant l'arbre de dépendances comme représentation. Afin de confirmer que ce choix est judicieux compte tenu de notre hypothèse sur le rôle capital des entités, nous calculons la distance entre chaque déclencheur événementiel et chaque entité. Nous présentons l'histogramme cumulé de ces distances en Figure 4.1.1. Les entités sont plus proches dans la représentation syntaxique que dans la forme de surface, ce qui confirme notre hypothèse. L'intérêt de cette représentation est plus concrètement illustrée à la Figure 4.1.2 présentant les dépendances syntaxiques de l'exemple précédent : bien que "*Attorney*" et "*fired*" soient à une distance de 9 dans la représentation de surface, ces deux mots sont voisins directs dans l'arbre de dépendance de la phrase, rendant de fait leur interaction plus aisément identifiable.

Cependant, cette représentation n'est plus une séquence mais un graphe acyclique où les nœuds et les arêtes sont respectivement constitués des mots et des dépendances syntaxiques entre un gouverneur et un gouverné. Il n'est donc plus possible d'utiliser les modèles convolutifs et récurrents classiques et nécessaire de se tourner vers d'autres modèles spécifiquement conçus pour opérer sur des structures de graphes.

Les modèles récursifs (Goller et Küchler, 1996) sont une généralisation des modèles

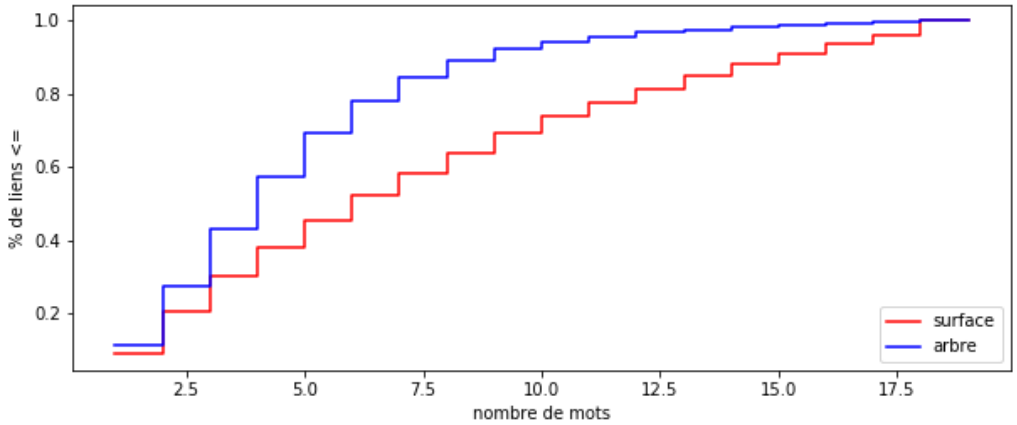

FIGURE 4.1.1 – Histogramme cumulé des distances entre mentions d'entités et déclencheurs sur le jeu de données $TAC15_{train}$. Les entités utilisées sont celles détectées automatiquement par l'outil Stanford CoreNLP. Pour chaque couple (déclencheur-mention d'entité) au sein d'une phrase, nous calculons la distance dans la forme de surface et la taille du chemin de dépendances syntaxiques (en nombre de mots). Les distances médianes dans l'arbre et la forme de surface sont respectivement 5 et 8.

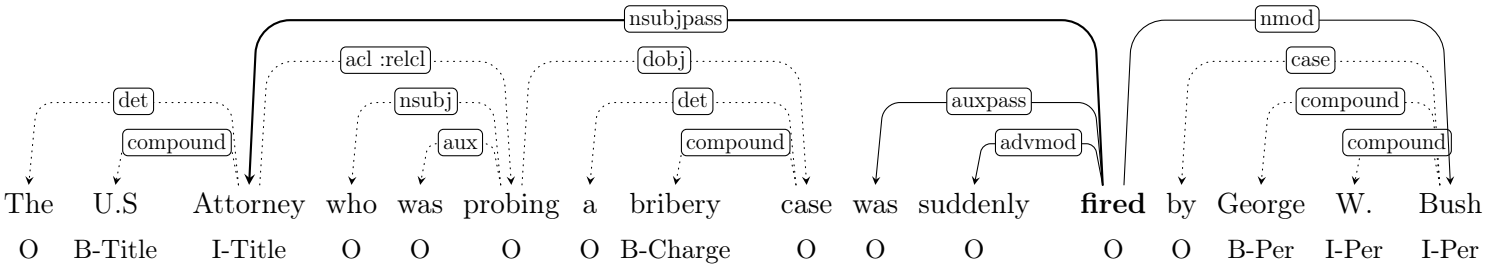

FIGURE 4.1.2 – Représentation des dépendances syntaxiques et des annotations IOB [1] des entités de la phrase d'exemple. Les arcs représentent les dépendances du gouverneur au gouverné. Les types d'entités "Title","Charge" et "Per" identifient respectivement des mentions de professions, accusations et personnes.

récurrents opérant sur des graphes dirigés acycliques et pouvant donc pleinement exploiter la structure des dépendances syntaxiques. Ces modèles ont notamment été utilisés par Socher (2014) pour différentes tâches de classification structurée telle que l'analyse syntaxique, la prédiction de sentiment ou la détection de paraphrases. Toutefois, à cause de la difficulté de leur apprentissage, ils n'ont pour l'instant jamais été utilisés en extraction d'événements. Récemment proposés pour l'exploitation de graphes, les modèles neuronaux reposant sur la convolution de graphe (Kipf et Welling, 2017) constituent une autre

1. L'annotation *Inside-Out-Beginning* (IOB) est un format d'annotation des tokens permettant de représenter des annotations multi-tokens en préfixant les étiquettes d'annotation en fonction de leur position dans la séquence annotée. Le premier token d'un groupe (ici, une entité) est préfixé par "B-", les suivants par "I-" et les tokens sans étiquette sont préfixés par "O".

solution pour tirer profit de cette représentation en dépendance. Ils permettent de plus, contrairement aux modèles récursifs, d'introduire des cycles dans le graphe. Tout comme la convolution classique, la convolution de graphe consiste à appliquer une même opération sur le voisinage de chacun des mots de la phrase, indépendamment de leur position dans celle-ci. Cependant, le modèle n'opérant pas sur une séquence, la transformation appliquée n'est pas spécifique à la position du mot dans la fenêtre, comme c'est le cas pour une convolution classique. Il est en revanche possible de conditionner la transformation utilisée au type d'arête reliant le mot courant à un voisin. De plus, à la différence de la convolution textuelle classique dont le champ récepteur est de taille fixe, la convolution de graphe considère pour chaque nœud l'ensemble de ses voisins directs pour produire sa représentation de sortie. Il est toutefois possible d'accéder à des informations distances par propagation de proche en proche comme nous l'illustrons dans la Figure 4.1.3.

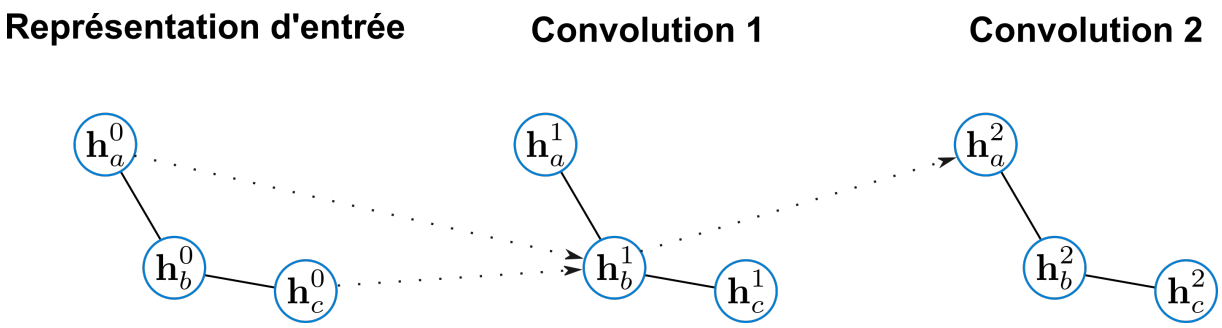

FIGURE 4.1.3 – Illustration de la propagation d'informations à l'aide d'une convolution de graphe. Les flèches pointillées indiquent les dépendances prises en compte pour la convolution d'un nœud. Nous ne représentons ici que les dépendances utilisées dans l'exemple ci-dessous.

La notation h_i^k spécifie la représentation du nœud i à la couche k. Dans la première partie de la figure, $k = 0$, les représentations des nœuds sont leur représentation d'entrée, qui peuvent par exemple être des plongements de mots. La représentation h_b^1 du nœud b à la couche 1 est obtenue en multipliant par une même matrice de poids les représentations des voisins de ce nœud dans la couche précédente, h_a^0 et h_c^0, puis en agrégeant ces représentations. On calcule de la même manière les représentations des autres nœuds. De manière analogue, à la couche $k = 2$, on obtient la représentation h_a^2 du nœud a en multipliant par une matrice de poids la représentation à la couche précédente de son seul voisin, h_b^1. Celle-ci étant conditionnée par h_c^0, la représentation finale du nœud a peut tenir compte du nœud c bien qu'ils ne soient pas voisins directs. Ainsi, avec K couches, le modèle est capable d'exploiter des dépendances de longueur K.

Le modèle de convolution de graphe proposé par Nguyen et Grishman (2018) pour l'extraction d'événements ayant obtenu de bonnes performances en détection d'événements, nous considérerons par la suite cette architecture. Ce modèle est constitué de quatre composantes principales : une couche de plongement associant une représentation vectorielle à chaque mot, un BiLSTM appliqué sur la forme de surface permettant une contextualisation des représentations d'entrée, puis plusieurs couches de convolution de graphe opérant sur ces représentations et enfin une couche de pooling permettant d'agréger les représentations avant la couche de classification. Nous présentons ici plus en détail ces composants.

4.1.2 Représentation des entrées

Comme pour les modèles convolutifs des précédents chapitres, la tâche de détection d'événements est envisagée sous la forme d'une classification multiclasse pour chaque mot de la phrase. Pour chaque candidat w_t, nous créons donc une représentation de la phrase $w = (w_1, w_2, \ldots, w_n)$ dans laquelle il apparaît. Pour ce faire, pour chaque mot w_i, sa représentation x_i est obtenue en concaténant les représentations vectorielles réelles suivantes :

— **plongement de mot** cette représentation encode les propriétés syntaxiques et sémantiques du mot w_i et est initialisée à partir de représentations générées sur la base d'un large corpus non annoté.

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

— **plongement d'entités** tout comme pour les plongements de position, chaque type d'entité (dont le type "None") 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.

Une fois cette représentation produite pour chaque mot, leur concaténation fournit la séquence X :

$$X = (x_0, x_1, \ldots, x_n) \tag{4.1}$$

Celle-ci constitue une représentation de la phrase, centrée sur le mot cible grâce aux plongements de position. Elle est ensuite utilisée en entrée d'un modèle BiLSTM permettant de produire une représentation contextualisée de chaque mot de la séquence.

Nous appliquons également un analyseur syntaxique à la phrase afin d'identifier les dépendances syntaxiques qui serviront à produire le graphe présenté dans la section suivante.

4.1.3 Convolution de graphe

À partir d'une phrase w de n mots w_1, w_2,...w_n, le graphe $G = \{V, E\}$ est constitué de l'ensemble des nœuds $V = \{w_1, w_2, \ldots, w_n\}$ et de l'ensemble des arêtes E. 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 :

$$h_u^{k+1} = \sigma\left(\sum_{v \in N(u)} W_{L(u,v)}^k h_v^k + b_{L(u,v)}^k\right) \tag{4.2}$$

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, nous employons comme Nguyen et Grishman (2018) la pondération des voisins. Pour la première couche du graphe, la représentation h_u^0 est la représentation contextualisée du mot x_u produite par le BiLSTM.

4.1.4 Pooling

Une fois produite la séquence des représentations vectorielles $h^k_{w_1}, h^k_{w_2}, \ldots h^k_{w_n}$ 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 et 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 :

$$p_t = maxpool(\{h^K_{w_t}\} \cup \{H^K_{w_i} : 1 \leq i \leq n, e_i \neq None\}) \tag{4.3}$$

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.

4.2 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. À notre connaissance, seulement deux autres études se sont portées sur l'intégration de ce contexte distant. 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. Nous présentons donc dans cette section notre deuxième contribution consistant en la production et l'intégration d'une telle représentation au modèle local présenté dans la section précédente. Pour ce faire, nous analysons tout d'abord les propositions existantes de représentations du contexte inter-phrastique et leurs limites potentielles et proposons une présentation générale de notre contribution. Dans la suite de la section, nous présentons successivement les étapes de sélection du contexte inter-phrastique, de production d'une représentation de ce contexte et enfin d'intégration de cette représentation au modèle local.

4.2.1 Problématique

Comme nous venons de le dire, peu d'études se sont intéressées à l'intégration des informations du contexte inter-phrastique à un modèle neuronal local. Duan *et al.* (2017), auxquels nous nous sommes comparé dans le chapitre 2, proposent d'intégrer un plongement

du document courant produit à l'aide de doc2vec à un modèle récurrent bidirectionnel. Cette représentation peut être limitante à deux égards. D'une part, cette représentation est produite de manière non-supervisée et n'est donc pas spécifique à la tâche. Nous supposons qu'une représentation spécifique est préférable afin d'extraire les traits les plus à même de contribuer à résoudre la tâche d'extraction d'événements. Par ailleurs, la représentation proposée par les auteurs est la même pour l'ensemble du document. Dans des cas d'ambiguïtés relativement simples, notamment entre deux types d'événements dont un seul est effectivement présent dans le document (p. ex. "*fired*" dans un document parlant uniquement de licenciements), une représentation unique du contexte peut suffire. Mais dans des cas plus complexes, notamment en présence d'ambiguïtés entre deux types d'événements présents dans le document ou entre deux types d'événements proches (*charge* - *convict* - *sentence*), il peut être nécessaire d'exploiter le contexte spécifique de la mention ambiguë.

La méthode proposée par Zhao *et al.* (2018) consiste à produire une représentation du document à l'aide d'un modèle hiérarchique de document : l'application d'un BiLSTM aux représentations vectorielles des mots puis leur agrégation par un mécanisme d'attention produisent une représentation des phrases. Ces représentations sont ensuite traitées par une seconde couche analogue opérant sur les représentations de phrases pour produire une représentation du document. Cette représentation est alors concaténée aux plongements de mots et d'entités pour constituer l'entrée d'un perceptron multicouche. L'entraînement joint du modèle hiérarchique de document et du modèle local garantit la spécialisation de la représentation sur la tâche d'extraction d'événements. Cette spécialisation est par ailleurs renforcée par l'application d'une fonction de coût au vecteur d'attention : durant l'apprentissage, le modèle est pénalisé lorsque l'attention sur les mots ne sélectionne pas les déclencheurs ou lorsque l'attention sur les phrases ne le fait pas pour les phrases en contenant. Cependant cette représentation est encore une fois la même pour l'ensemble du document. De plus, pour des documents longs, l'application du modèle hiérarchique à l'ensemble du texte peut s'avérer coûteux voire impossible.

Un principe limitant le contexte, pour une phrase cible, aux phrases utiles parmi

l'ensemble des phrases du document permettrait de restreindre les ressources de calcul nécessaires à l'entraînement du modèle. 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. Prenons ici comme exemple les deux phrases suivantes provenant d'un même document du jeu de données ACE 2005 :

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

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

Dans la seconde phrase, "pow-wow" est déclencheur d'un événement *Meet* mais ce mot est particulièrement ambigu. Cependant, l'identification de la coréférence entre les deux mentions "Saint Petersburg" permet de relier cette phrase à la première dans laquelle la présence de l'événement est évidente. Guidés par cette intuition, 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.

4.2.2 Sélection des phrases de contexte

À partir d'une phrase $w^j = (w_1^j, w_2^j, \ldots, w_n^j)$, pour chaque mot w_i^j tel que $e_i^j \neq$ "None", $E(e_i^j)$ désigne l'entité correspondant à la mention e_i^j. Pour deux phrases w^j et w^k, $links(w^j, w^k)$ engendre l'ensemble des liens d'entités entre les deux phrases.

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

Le contexte d'une phrase w^j est alors donné par $Links(w^j, min_{links})$. Cette fonction produit l'ensemble des tuples associant une phrase de contexte et les liens qu'elle entretient avec la phrase cible. Afin d'éviter que des entités peu informatives ne perturbent l'extraction, nous introduisons un hyperparamètre min_{links} contrôlant le nombre de liens distincts minimum qu'une phrase de contexte doit entretenir avec la phrase cible pour être considérée.

$$Links(w^j, min_{links}) = \left\{ \left(w^k, links(w^j, w^k) \right) \ : \ j \neq k \text{ et } |links(w^j, w^k)| > min_{links} \right\} \tag{4.4}$$

ou $|links(w^j, w^k)|$ est le cardinal de $links(w^j, w^k)$.

4.2.3 Extraction des représentations du contexte

Nous souhaitons produire des représentations spécifiques pour chaque entité e_i^jde la phrase cible w^j. Nous définissons donc *ent-links*(w^j, w^k, i) qui, pour une mention d'entité dans la phrase cible et une phrase de contexte, renvoie les positions des mentions en coréférence dans la phrase de contexte.

$$ent\text{-}links(w^j, w^k, i)) = \{m : E(e_i^j) = E(e_m^k)\} \tag{4.5}$$

De manière analogue, on définit $Ent\text{-}Links(w^j, min_{links}, i)$ qui, pour une mention d'entité de la phrase cible et un nombre de liens minimum, produit l'ensemble des mentions

d'entités en coréférence dans des phrases ayant suffisamment de liens avec la phrase cible.

$$Ent\text{-}Links(w^j, min_{links}, i) = \left\{ (w^k, l) \ : w^k \in Links(w^j, min_{links}) \right. \\ \left. \text{et } l \in ent\text{-}links(w^j, w^k, i) \right\} \tag{4.6}$$

Pour chacune de ces paires (w^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 4.1.2 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.

Nous appliquons ensuite un modèle récurrent bidirectionnel $BiLSTM_{context}$ à cette représentation. Deux méthodes d'extraction sont possibles : le mode "Finale" (eq. 4.7) consiste à concaténer les représentations finales des deux modèles récurrents tandis que le mode "Mention" (eq. 4.8) extrait les représentations à l'emplacement de l'entité.

$$\textbf{Finale} : h_{\textbf{context}}(w^{k,l}) = [h_{\text{forward}}(x_n^{k,l}); h_{\text{backward}}(x_1^{k,l})] \tag{4.7}$$

$$\textbf{Mention} : h_{\textbf{context}}(w^{k,l}) = [h_{\text{forward}}(x_l^{k,l}); h_{\text{backward}}(x_l^{k,l})] \tag{4.8}$$

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

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

4.2.4 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 4.2.1.

Que ce soit pour une intégration au niveau des plongements ou du graphe, la représentation attendue est un vecteur. Nous agrégeons donc l'ensemble des vecteurs obtenus

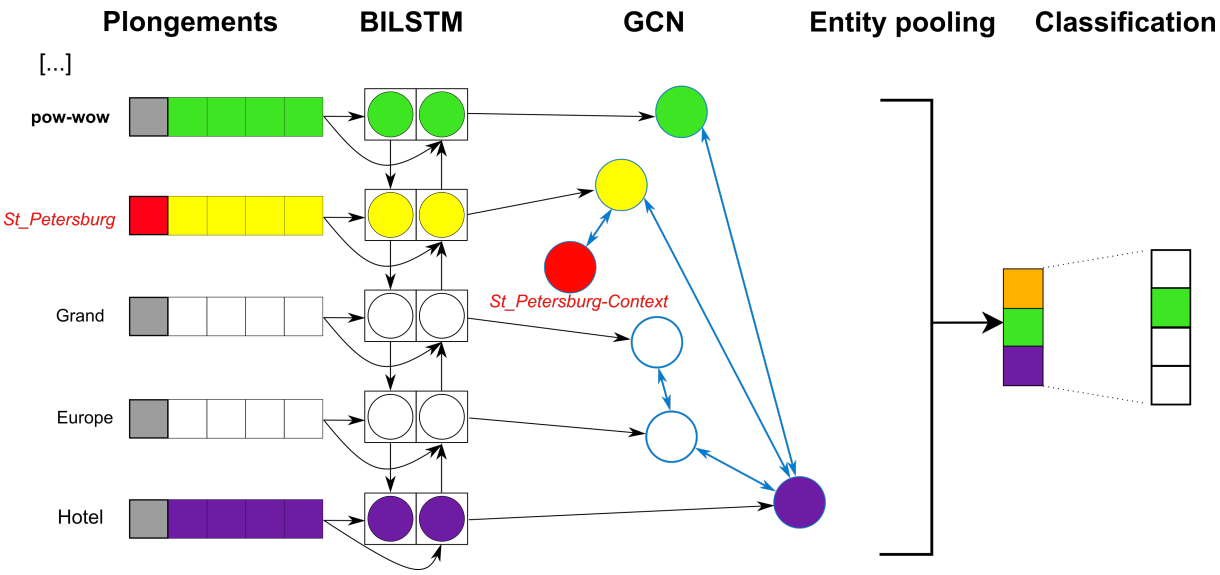

FIGURE 4.2.1 – Possibilités d'intégration d'une représentation contextuelle d'entité au sein d'un modèle de convolution de graphe. Cette représentation (en rouge) peut être intégrée en entrée du modèle ou dans le graphe par l'introduction d'un nouveau nœud connecté à la mention locale par un lien de type "contexte". Le changement de couleur du jaune au orange explicite l'influence du contexte sur la représentation finale.

via une étape de maxpooling permettant d'obtenir le vecteur de contexte [2] :

$$context_i = maxpool(\{h_{\mathbf{context}}(w^{k,l}) \ : \ (w^k, l) \in Ent\text{-}Links(w^j, min_{links}, i)\} \tag{4.9}$$

Pour l'intégration au niveau des nœuds, nous modifions donc le graphe $G = \{V, E\}$ en ajoutant un nœud $w_{i'}$ dans V, dont la représentation initiale est $h^0_{i'} = context_i$. 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 V. 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 $c_{default}$ sera modifiée durant l'apprentissage. Le vecteur de contexte 4.9 est alors généralisé à l'ensemble des mots de la phrase en

2. Par souci de concision et de cohérence avec la notation utilisée en section 4.1.2, nous n'utilisons pas l'index de phrase cible j en exposant de $context_i$.

introduisant :

$$c_i = \begin{cases} context_i & \text{si } |Ent\text{-}Links(w^j, min_{links}, i)| > 0 \\ c_{default} & \text{sinon} \end{cases} \tag{4.10}$$

En y introduisant le vecteur 4.10, nous redéfinissons la séquence d'entrée 4.1 ainsi :

$$X = ([x_0, c_0], [x_1, c_1], \ldots, [x_n, c_n]) \tag{4.11}$$

4.3 Données

4.3.1 Prétraitements

Nous nous évaluons encore ici sur les différents jeux de données annotés au format Rich ERE, ceci nous permettant de nous comparer aux performances originelles du modèle de graphe présenté par Nguyen et Grishman (2018) sur $\text{TAC15}_{\text{test}}$ ainsi qu'à nos modèles convolutifs sur $\text{TAC17}_{\text{test}}$. 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.

4.3.2 Jeux de données

Dans les expériences présentées dans la suite, nous utilisons plusieurs partitionnements spécifiques des jeux de données afin d'étudier l'influence de différents paramètres sur les

performances. Dans les trois premiers partitionnements, le jeu de données $\text{TAC15}_{\text{train}}$ est partagé entre les corpus d'entraînement et de validation. C'est pourquoi ces partitionnements spécifient à la fois le jeu d'entraînement et de validation. Lorsque nous ferons plus tard référence à ces partitionnements, nous spécifierons ainsi ces deux ensembles. Pour les autres partitionnements ne définissant qu'un jeu d'entraînement, nous préciserons le jeu de validation utilisé.

— **deft/tac** Le jeu d'entraînement est constitué de 58 documents issus de TAC 2015 train et des 288 documents issus des jeux de données DEFT Rich ERE (R2 V2 et V2). L'ensemble de validation est composé des 100 documents restants de TAC 2015 train. Le corpus d'entraînement est donc hétérogène du point de vue des annotations, la principale différence étant la présence dans TAC de déclencheurs associés à deux types d'événements à la fois.

— **tac-train+** Afin de nous comparer au modèle original de Nguyen et Grishman (2018), n'utilisant que le jeu de données TAC 2015 pour l'entraînement, nous considérons ici une séparation entre données d'apprentissage et de validation privilégiant ce premier ensemble. 80% des documents de TAC 2015 train (125 documents) sont utilisés pour l'ensemble d'apprentissage et les 20% restants (33 documents) pour le jeu de validation.

— **tac-val+** À l'inverse du précédent jeu de données composite, ce partitionnement vise à étudier l'influence d'un plus grand jeu de validation. 100 documents de TAC 2015 train sont utilisés pour l'ensemble d'apprentissage et les 58 restants pour le jeu de validation.

— **deft/tac-full** Ce jeu d'entraînement est constitué afin de maximiser le nombre de données d'entraînement antérieure à l'édition 2016 de la campagne TAC, dans l'optique d'utiliser les documents issus de cette édition pour la validation et de s'évaluer sur l'édition 2017. Il est obtenu par fusion des jeux de données d'entraînement et de test TAC 2015 (360 documents) et de DEFT Rich Ere V2 et R2 V2 (288 documents) et contient 648 documents.

— **tac-full** Ce jeu d'entraînement est obtenu par fusion des jeux d'entraînement et de

test TAC 2015 et contient 360 documents.

4.3.3 Génération des exemples

Comme nous l'avons vu lors de la présentation de la convolution de graphe en section 4.1.3, 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. Nous appelons ce mode de filtrage "strict". Nous considérons également une alternative, le filtrage "assoupli" : ce filtrage conserve également les déterminants, prépositions et conjonctions. 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 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.

4.4 Expériences

4.4.1 Hyperparamètres

Nous nous appuyons sur la suite d'outils linguistiques Stanford CoreNLP pour réaliser l'extraction d'entités nommées, la résolution de coréférences et l'analyse en dépendances utilisée pour produire les graphes. Les dépendances considérées sont les dépendances "*Ba-*

sic 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. Le modèle est entraîné via SGD avec momentum avec des lots de 10 exemples. Concernant le paramètre min_{links} limitant le nombre de phrases de contexte considéré, nous avons déterminé lors d'expériences préliminaires que sa valeur optimale était 1. Les autres paramètres sont déterminés par optimisation bayésienne d'hyperparamètres à l'aide de la bibliothèque hyperopt [3], les configurations présentées étant sélectionnées à l'aide des performances en validation. Toutes les performances moyennes fournies sont calculées pour 10 reproductions avec les mêmes paramètres.

4.4.2 Étude des hyperparamètres du modèle

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

— **Filtrage des mots** *Strict* / *Assoupli* (section 4.3.3)

— **Extraction** *Finale*/*Mention* (section 4.2.3)

— **Intégration** *Plongement*/*Nœud* (section 4.2.4)

— **Pooling** *Cible*/*Syntaxique*/*Entité* (section 4.1.4)

Nous avons réalisé sur le jeu de données deft/tac 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 le filtrage *Strict*, 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 la Table

3. `https://github.com/hyperopt/hyperopt`

	$P_{moy.}$	$R_{moy.}$	$F_{moy.}$	F_{σ}	$F_{max.}$
C-GCN	63,39	**57,34**	**60,19**	0,20	**60,51**
Intégration - Nœud	**65,53**	55,40	59,96	0,38	60,39
Pooling - Entité	63,35	57,07	59,96	0,30	60,49
Extraction - Mention	63,11	57,07	59,86	0,44	60,40
Filtrage - Assoupli	64,21	56,08	59,79	0,26	60,27
Pooling - Cible	62,14	56,92	59,37	0,36	59,99

TABLE 4.4.1 – Performances en validation sur le jeu de données composite deft/tac en fonction des choix de modélisation, avec la précision, le rappel et la f-mesure moyennes ($P_{moy.}$, $R_{moy.}$, $F_{moy.}$), l'écart-type F_{σ} de $F_{moy.}$ et la f-mesure maximale ($F_{max.}$)

.

4.4.1 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 observerons 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 et Grishman (2018) qui obtient dans l'article d'origine des performances plus basses que le pooling des entités. Puisque nous utilisons un système de détection d'entités différent de celui utilisé par Nguyen et 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.

Concernant la génération des exemples, le modèle semble mieux opérer sur une représentation compressée de la phrase dans laquelle les mots clés sont plus proches de la cible à prédire que sur une représentation plus complète et nuancée. Le modèle obtient ainsi un

gain significatif lorsque nous utilisons le filtrage Strict plutôt que le filtrage Assoupli. 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.

4.4.3 Comparaison avec l'état de l'art

Nous comparons maintenant notre réimplémentation du modèle de graphe et notre proposition d'extension à l'implémentation originale ainsi qu'aux deux meilleurs modèles de la campagne d'évaluation TAC. La configuration présentée précédemment étant entraînée sur un ensemble de documents plus important que celui utilisé par les auteurs du $\text{GCN}_{\text{nguyen}}$, nous présentons également deux modèles développés, comme dans leur cas, uniquement sur le jeu de données $\text{TAC15}_{\text{train}}$. Nous utilisons plus spécifiquement la version tac-train+. On notera cependant que Nguyen et Grishman (2018) ont préalablement optimisé leur modèle sur ACE 2005 et ont donc fait usage de plus de données que le simple corpus utilisé pour l'entraînement. Afin de confirmer l'hypothèse qu'un contexte spécifique pour chaque exemple est préférable, nous entraînons également $\text{C-GCN}_{\text{generic}}$, exploitant l'ensemble des phrases du document en tant que contexte. Dans ce cas, le vecteur de position n'est pas utilisé pour les phrases de contexte et la représentation produite est utilisée comme plongement pour l'ensemble des mots de la phrase cible.

Les modèles présentés sont :

— **$\text{GCN}_{\text{nguyen}}$** Implémentation originelle du modèle de convolution de graphe, entraîné sur $\text{TAC15}_{\text{train}}$, sans jeu de validation.

— **RPI_BLENDER** Ce système proposé par Hong *et al.* (2015), ayant obtenu la première place lors de la campagne TAC 2015, est constitué d'un classifieur d'entropie maximale utilisant un ensemble de traits lexicaux, syntaxiques et de types d'entités, amélioré par l'introduction d'une procédure d'augmentation de données utilisant FrameNet. Une seconde passe de désambiguïsation fondée sur un modèle thématique (*topic model*) et la collecte automatique de documents similaires est ensuite appliquée. Le modèle est entraîné sur $\text{TAC15}_{\text{train}}$ ainsi que des documents

	$P_{moy.}$	$R_{moy.}$	$F_{moy.}$	F_{σ}	$F_{max.}$
$GCN_{tac\text{-}train+}$	71,26	48,46	57,49	1,55	58,3
$C\text{-}GCN_{tac\text{-}train+}$	72,07	48,41	57,91	0,7	57,3
$GCN_{deft/tac}$	78,48	46,96	58,73	0,82	59,1
$C\text{-}GCN_{generic}$	74,50	48,35	58,64	0,57	59,04
$C\text{-}GCN_{deft/tac}$	75,57	50,42	**60,47**	0,64	60,35
GCN_{nguyen}	70,3†	50,6†	-	-	58,8
RPI_BLENDER	75,23†	47,74†	-	-	58,41
LCC	73,95†	46,61†	-	-	57,18

TABLE 4.4.2 – Performances sur $TAC15_{test}$. † : valeurs maximales et non moyennes

externes pour la seconde passe.

— **LCC** Ce système (Monahan *et al.*, 2015), classé deuxième lors de la campagne d'évaluation TAC 2015, utilise un modèle de régression logistique exploitant des attributs lexicaux et contextuels. Les attributs lexicaux expriment des probabilités conditionnelles de la présence d'un type d'événement en fonction du mot ou du lemme et les attributs contextuels s'appuient sur une représentation de phrase produite par doc2vec (Le et Mikolov, 2014). En complément du corpus d'entraînement $TAC15_{train}$, le modèle exploite également plusieurs jeux de données additionnels lors de l'apprentissage.

— **$GCN_{deft/tac}$** Notre réimplémentation de GCN, entraînée sur deft/tac.

— **$C\text{-}GCN_{deft/tac}$** Notre modèle de contexte entraîné sur ces mêmes données.

— **$C\text{-}GCN_{generic}$** Ce modèle ne tient pas compte des liens de coréférence au sein du document et considère toutes les autres phrases au sein du contexte inter-phrastique. Il est également entraîné sur deft/tac.

— **$GCN_{tac\text{-}train+}$** Notre réimplémentation de GCN, entraînée sur tac-train+.

— **$C\text{-}GCN_{tac\text{-}train+}$** Notre modèle utilisant la représentation de contexte, également développé sur tac-train+.

Les résultats présentés dans la Table 4.4.2 confirment l'intérêt de notre proposition en présence de suffisamment de données d'apprentissage et de validation. En effet, l'introduction de notre représentation de contexte apporte un gain de 1,74 points en F-mesure en utilisant le jeu de données deft/tac ($p < 0,0001$) et permet de dépasser le modèle

$\text{GCN}_{\text{nguyen}}$ détenant jusqu'alors la meilleure performance sur $\text{TAC15}_{\text{test}}$, ainsi que les deux meilleurs modèles de la campagne, utilisant eux aussi plus de données que le simple corpus $\text{TAC15}_{\text{train}}$ et ayant recours à des ensembles larges d'attributs définis manuellement.

Nous constatons également que l'intégration de la représentation de contexte dans $\text{C-GCN}_{\text{generic}}$ 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. Sur tac-train+, nous observons également un gain en moyenne avec le modèle C-GCN par rapport au modèle GCN, bien que ce gain soit plus faible. Cependant, dans ces conditions, ni notre modèle local ni son extension ne permettent d'atteindre les performances de l'implémentation d'origine. De plus, malgré l'apport en moyenne de notre contribution, dans ces conditions, la situation est inversée pour les performances maximales[4]. Ce constat est à mettre en lien avec la variance élevée du modèle local, qu'on observe également pour le jeu de données deft/tac mais qui semble ici amplifiée par la taille restreinte du jeu de validation (33 documents). Nous produisons en Figures 4.4.1 et 4.4.2 une analyse par classe des gains de $\text{C-GCN}_{\text{deft/tac}}$ par rapport au modèle local respectivement non normalisée et normalisée. La normalisation consiste à multiplier pour chaque classe la taille de la variation par le rapport entre le cardinal de cette classe et celui de la classe ayant le plus d'événements.

4. Nous rappelons ici que la performance *max* affichée correspond à la performance en test du meilleur modèle en validation, ce qui explique le fait que la performance *max* du modèle $\text{C-GCN}_{\text{tac-train+}}$ soit inférieure à sa performance moyenne.

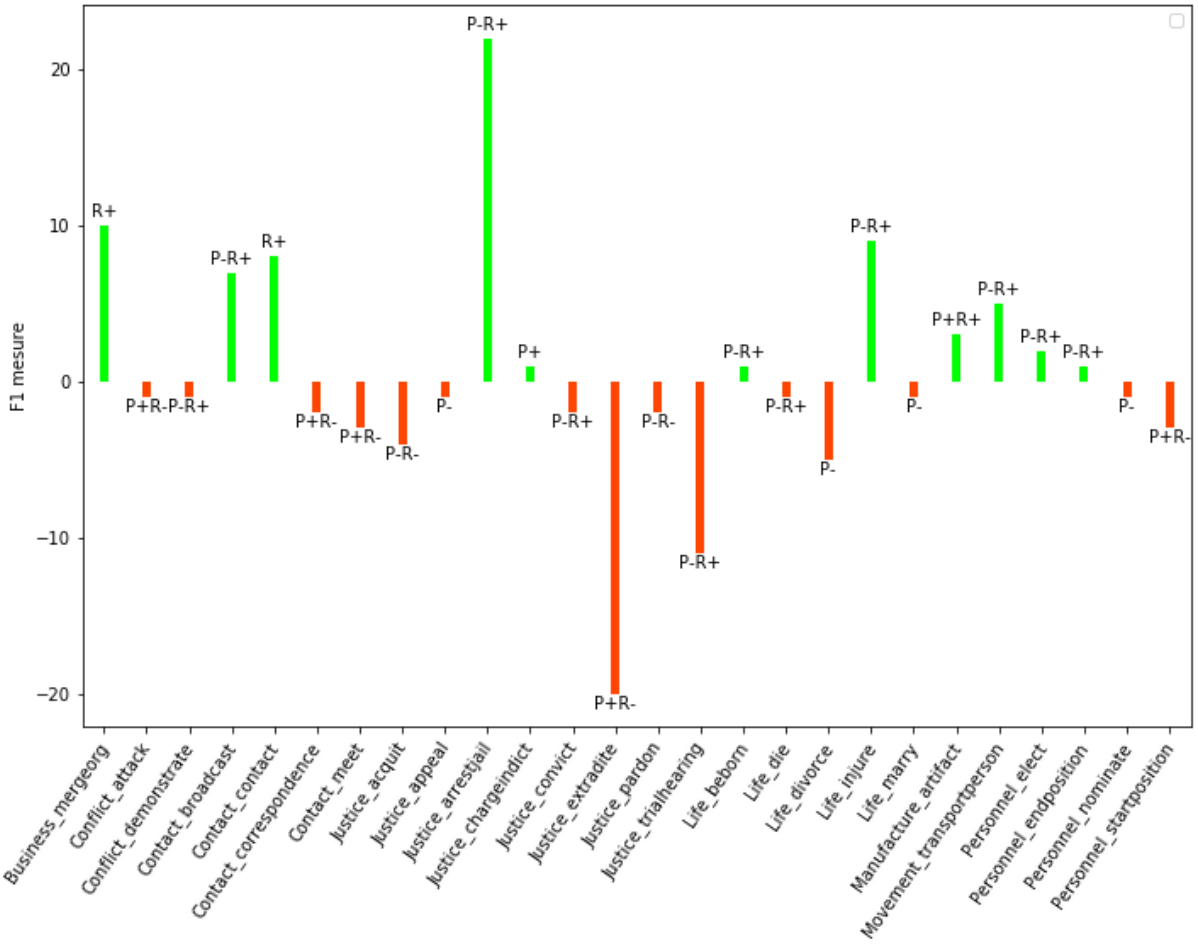

FIGURE 4.4.1 – Variation des F-mesures par classe entre GCN et C-GCN sur $\text{TAC15}_{\text{test}}$. Seules les classes avec une variation de performance sont présentées. Les variations en **P**récision et en **R**appel sont indiquées au-dessus de la barre, "-" pour une réduction et "+" une augmentation par rapport au modèle local.

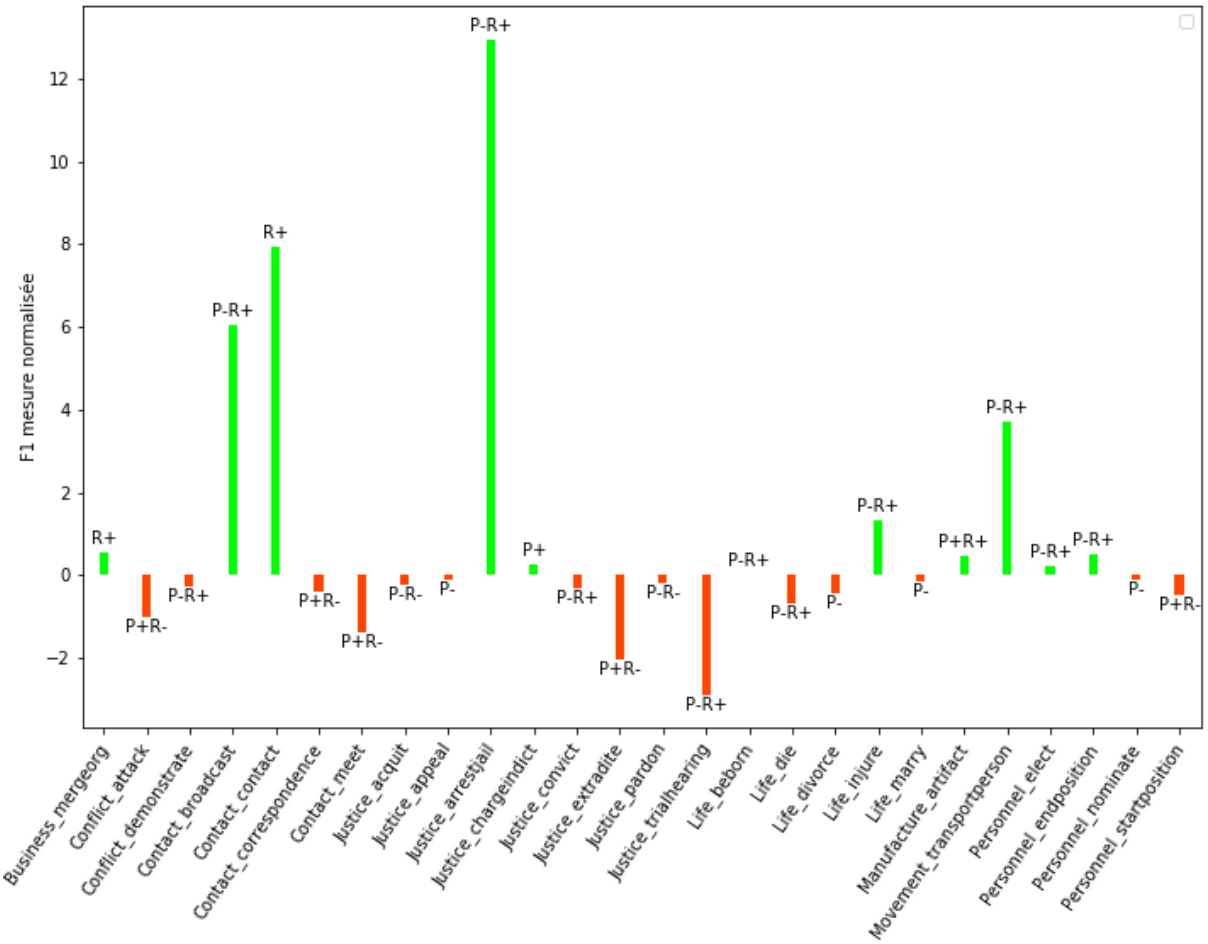

FIGURE 4.4.2 – Variation normalisée des F-mesures par classe entre GCN et C-GCN sur $\text{TAC15}_{\text{test}}$. Les variations sont normalisées par le nombre d'événements de chaque classe.

	nb docs						
modèle	train	valid	$P_{moy.}$	$R_{moy.}$	$F_{moy.}$	F_{σ}	$F_{max.}$
$GCN_{deft/tac}$	346 (58)	100	78,48	46,96	58,73	0,82	59,1
C-$GCN_{deft/tac}$			75,57	50,42	**60,47**	0,64	60,35
$GCN_{tac\text{-}train+}$	125	33	71,26	48,46	57,49	1,55	58,3
C-$GCN_{tac\text{-}train+}$			72,07	48,41	**57,91**	0,70	57,3
$GCN_{tac\text{-}val+}$	100	58	69,57	45,92	55,28	0,65	55,38
C-$GCN_{tac\text{-}val+}$			66,86	47,25	55,30	0,59	54,78

TABLE 4.4.3 – Influence de la taille des jeux d'apprentissage et de validation sur les performances sur $TAC15_{test}$. Le nombre entre parenthèses indique que 58 des documents d'apprentissage proviennent de $TAC15_{train}$.

Nous observons tout d'abord que les variations ne sont pas présentes pour l'ensemble des classes, seules 26 sur 38 affichant des variations. Parmi celles-ci, nous observons une augmentation du rappel dans plus de la moitié des cas (14) et une réduction de la précision dans 17 cas. Par ailleurs, nous constatons une augmentation des gains et une diminution des pertes entre les deux figures, indiquant que le gain de notre modèle est réalisé sur les classes dominantes au détriment des classes peu présentes.

Afin de mieux comprendre l'influence de la taille du jeu de validation mis en lumière par l'instabilité des modèles entraînés sur tac-train+, nous réalisons une expérience annexe dont les résultats sont présentés en Table 4.4.3. Nous comparons ici en test ces performances à celles obtenues en utilisant la version alternative de $TAC15_{train}$, tac-val+. Suivant la convention employée dans la table précédente, ces modèles sont suffixés par le jeu de données utilisé. Cette expérience ne permet toutefois pas d'isoler l'influence de la taille du jeu de validation, l'augmentation de celui-ci correspondant nécessairement à une diminution de l'ensemble d'apprentissage. Comme attendu, l'augmentation de la taille du jeu de validation dans tac-val+ permet une réduction de la variance des performances en test du modèle local. Nous constatons néanmoins une baisse importante des performances moyennes des deux modèles, à mettre en lien avec la réduction de 25 documents du jeu d'entraînement.

Nous remarquons également que la réponse du modèle local à l'augmentation du volume de données d'entraînement sature plus vite que pour le modèle C-GCN. Les deux

modèles voient effectivement leurs performances augmenter de manière comparable (2,21 et 2,61) lors du passage entre tac-val+ et tac-train+ (25 documents). Mais l'augmentation de 221 documents en passant à deft/tac ne permet d'augmenter le modèle local que de 1,24 point tandis que le modèle global bénéficie d'un gain de 2,56 points. Nous en concluons que le C-GCN nécessite un volume de données plus important pour tirer pleinement partie du BiLSTM de contexte employé. Cette conclusion est en accord avec l'analyse du gain par classe présentée précédemment où les pertes proviennent essentiellement des classes peu présentes et inversement pour les gains.

Nous signalons ici que ce jeu de données est plus hétérogène et distinct du jeu de test puisqu'il contient une majorité de documents issus des corpus DEFT Rich ERE. Or, ces corpus n'ont pas été annotés de la même manière, la différence la plus notoire étant l'absence de déclencheurs hybrides (i.e annotés avec plusieurs types d'événements). Le gain substantiel observé sur le modèle C-GCN semble donc indiquer que l'augmentation du volume de données d'entraînement prime sur leur similarité strict au jeu de test.

Nous évaluons maintenant notre modèle sur le jeu de données $\text{TAC17}_{\text{test}}$ afin de le comparer au modèle convolutif proposé au chapitre 3. Les modèles sont entraînés sur deft/tac-full et le jeu de test $\text{TAC16}_{\text{test}}$ est utilisé pour la validation. Pour rappel, les jeux de validation et de test utilisés ne sont annotés que pour 18 des 38 sous-types d'événements présents dans le corpus d'entraînement. Comme pour notre précédent modèle, nous entraînons toujours notre modèle sur les 38 sous-types (ainsi que les types hybrides) mais nous remplaçons les sous-types absents par la classe NULLE lors de la phrase de prédiction. Ici aussi, les configurations présentées sont celles obtenues via l'optimisation des différents hyperparamètres du modèle et de l'optimiseur.

Nous comparons nos modèles (GCN et C-GCN) aux 3 meilleurs modèles de la campagne d'évaluation TAC Event Nugget 2017 ainsi qu'à notre précédente contribution. Les modèles de la campagne ayant déjà été décrits dans le chapitre 2, nous les présentons succinctement et renvoyons le lecteur à ce chapitre pour plus de détails.

1. **Méthode d'ensemble BiLSTM CRF** Jiang *et al.* (2017) utilisent un ensemble de 10 modèles BiLSTM combinés par une stratégie de vote à un modèle CRF.

	$P_{moy.}$	$R_{moy.}$	$F_{moy.}$	F_{σ}	$F_{max.}$
CNN (Kodelja)	54,23†	46,59†	-	-	50,14
BILSTM à large marge (Makarov)‡	52,16†	48,71†	-	-	50,37
BILSTM CRF (Jiang)‡	56,83†	55,57†	-	-	56,19
$CNN_{doc\text{-}softmax}$	53.12	49.61	51,3	0,22	51,58
$GCN_{deft/tac\text{-}full}$	62,17	49,12	54,85	0,35	55,05
$C\text{-}GCN_{deft/tac\text{-}full}$	61,81	49,75	55,11	0,53	55,35

TABLE 4.4.4 – Comparaison avec l'état de l'art sur le jeu de données $TAC17_{test}$. Nos modèles sont entraînés sur deft/tac. † : valeurs maximales et non moyennes, ‡ : modèles d'ensemble.

2. **BiLSTM à large marge** Makarov et Clematide (2017) utilisent un ensemble de 5 BiLSTMs dotés d'un objectif à large marge pénalisant plus fortement les faux négatifs.
3. **Modèle convolutif** ce modèle est le CNN présenté dans le chapitre 2.
4. **$CNN_{doc\text{-}softmax}$** Notre précédente contribution, obtenue par la concaténation d'une représentation vectorielle du document aux descripteurs d'un modèle convolutif en entrée de la couche de prédiction. Cette représentation est produite par l'agrégation au niveau du document des prédictions d'un premier modèle convolutif.
5. **GCN** Notre implémentation du modèle de convolution de graphe proposée par Nguyen et Grishman (2018).
6. **C-GCN** Notre proposition d'extension, reposant sur l'extraction de représentations des phrases du contexte pour enrichir les plongements des entités de la phrase à prédire.

Les configurations des modèles GCN et C-GCN sont encore une fois obtenues par optimisation bayésienne. Ici, les deux meilleurs modèles utilisent le pooling *cible* au lieu du pooling *syntaxique* comme précédemment. Les résultats présentés montrent toujours un gain du modèle C-GCN sur le modèle local GCN. Cependant, le gain obtenu par l'introduction de notre représentation est plus faible que sur $TAC15_{test}$ et non significatif. On peut donc se questionner sur l'explication de cette diminution. L'analyse des divergences entre les différentes éditions de la campagne d'évaluation TAC permet d'identifier un élément de réponse. Premièrement, contrairement à l'édition 2015, les documents des

éditions 2016 et 2017 étaient communs à l'ensemble des tâches d'évaluation. Ils respectent donc les contraintes de la tâche de découverte d'entités qui impose la présence d'entités ambiguës dans les documents : "topics must have the potential to produce documents with ambiguous entities, including synonymous entities (different entities referenced by matching strings), polysemous entities (entities referenced by a variety of strings), and entities referenced only by nominal mentions". Bien que la présence d'entités ambiguës dans les autres corpus ne fasse aucun doute, cette contrainte supplémentaire ne peut que renforcer le phénomène. Le fait que le pooling cible ait été ici privilégié tant par le modèle local que son extension va également dans ce sens. Notre modèle reposant spécifiquement sur la bonne identification des entités et de leurs coréférences pour opérer, il est évident que ce phénomène influence négativement notre proposition.

Concernant la comparaison aux modèles de l'état de l'art, nos modèles obtiennent des performances supérieures à notre précédent modèle et relativement proche du système de Jiang *et al.* (2017), bien qu'inférieures. Il est cependant important de noter que ce système est un modèle d'ensemble composé de 10 BiLSTM et d'un CRF. Notre modèle global obtient ainsi les meilleures performances pour un modèle simple sur le jeu de données $\text{TAC17}_{\text{test}}$, bien que ces performances soient ici essentiellement dues à l'apport du modèle local. Les Figures 4.4.3 et 4.4.4 détaillent les performances par classes, respectivement sans normalisation et avec la même normalisation qu'à la Figure 4.4.2.

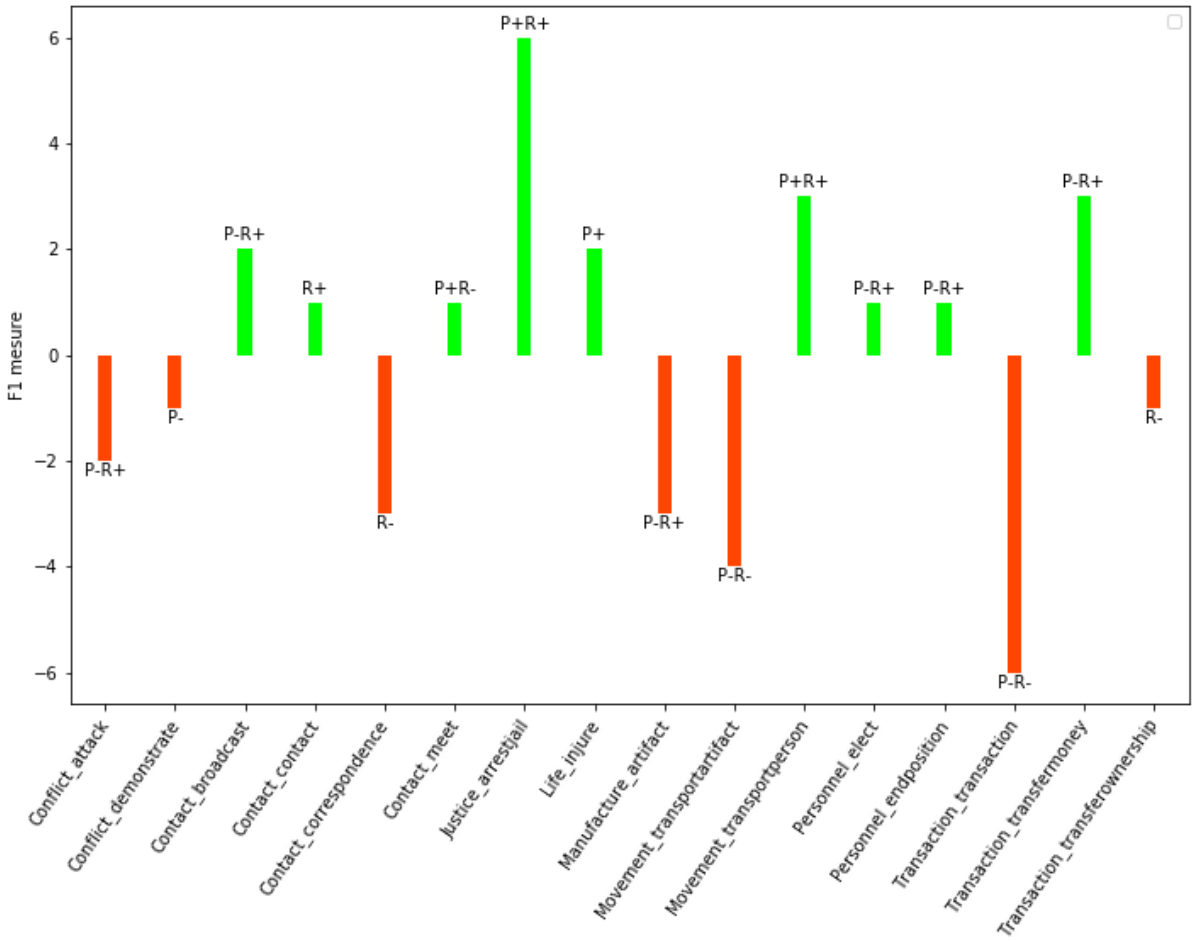

FIGURE 4.4.3 – Variation des F-mesures par classe entre GCN et C-GCN sur $\text{TAC17}_{\text{test}}$. Seules les classes avec une variation de performance sont présentées. Les variations en **P**récision et en **R**appel sont indiquées au-dessus de la barre, "-" pour une réduction et "+" une augmentation par rapport au modèle local.

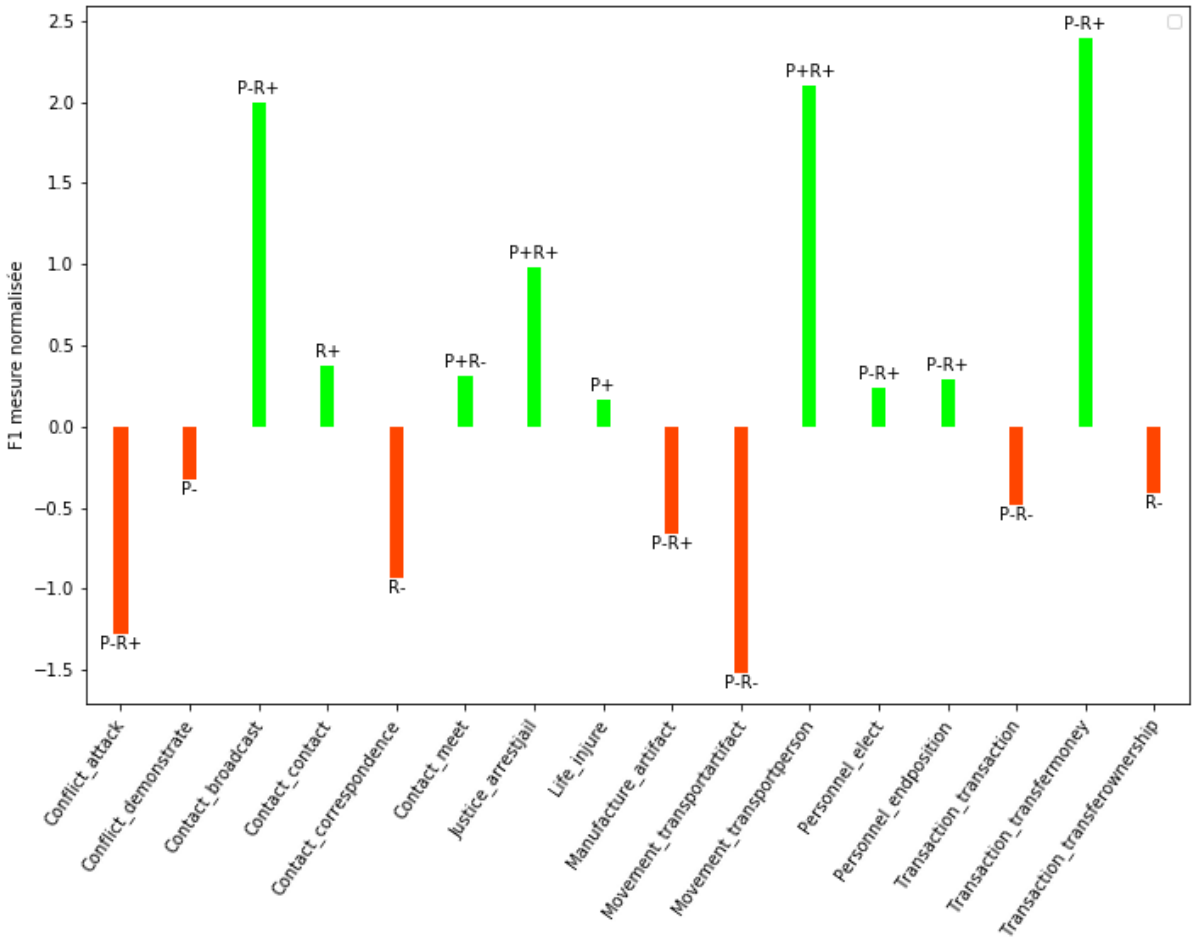

FIGURE 4.4.4 – Variation normalisée des F-mesures par classe entre GCN et C-GCN sur $\text{TAC17}_{\text{test}}$. Les variations sont normalisées par le nombre d'événements de chaque classe.

	$P_{moy.}$	$R_{moy.}$	$F_{moy.}$	F_{σ}	$F_{max.}$
$GCN_{deft/tac\text{-}full}$	62,17	49,12	54,85	0,35	55,05
$C\text{-}GCN_{deft/tac\text{-}full}$	61,81	49,75	55,11	0,53	55,35
$GCN_{tac\text{-}full}$	62,93	46,09	53,20	0,74	53,61
$C\text{-}GCN_{tac\text{-}full}$	61,51	46,82	53,10	0,44	53,12

TABLE 4.4.5 – Comparaison des performances de nos modèles sur le jeu de test $TAC17_{test}$ en fonction des jeux d'entraînement.

Afin de mieux analyser la réaction de notre modèle à l'hétérogénéité des données d'entraînement et à leur volume, nous présentons ici une expérience complémentaire. Nous entraînons cette fois les architectures GCN et C-GCN sur tac-full, le jeu d'entraînement obtenu en supprimant les jeux de données DEFT Rich ERE. Il s'agit donc d'un jeu d'entraînement dont la taille est divisée par deux mais qui est plus similaire aux documents d'évaluation. Ce jeu de données a néanmoins une taille comparable à deft/tac, qui obtient les meilleures performances sur $TAC15_{test}$. Il est donc possible que contrairement aux résultats présentés en Table 4.4.3, l'augmentation qualitative prime sur la perte quantitative. Malheureusement, ce n'est pas ce qu'on observe en Table 4.4.5 : comme précédemment, la diminution du volume de données influence négativement les deux jeux de données et efface cette fois complètement le gain de notre contribution. Les deux configurations tac-full n'ayant pas été obtenues par optimisation bayésienne mais simplement en entraînant sur ces données les configurations optimisées pour deft/tac-full, il semble que le modèle C-GCN soit plus dépendant d'un paramétrage précis de sa configuration.

4.5 Conclusion

Nous avons proposé dans ce chapitre 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. Nous avons réalisé sur TAC 2015 une évaluation des différents choix de modélisation puis des performances en test. Cette évaluation per-

met sur le jeu de test $\text{TAC15}_{\text{test}}$ d'obtenir un gain significatif par rapport au modèle de graphe nous servant de base. Les performances obtenues sont par ailleurs les meilleures obtenues sur ce jeu de données, confirmant l'intérêt de notre méthode.

L'évaluation de notre modèle sur le jeu de test $\text{TAC17}_{\text{test}}$ nous permet d'obtenir les meilleures performances pour un modèle simple sur ce jeu de données. Nous observons toujours un gain par rapport à notre modèle local, bien que ce gain soit sensiblement inférieur à celui obtenu sur le précédent jeu de données. Cette perte de performance semble provenir d'une plus grande ambiguïté des mentions d'entités sur ce nouveau jeu de données. Par ailleurs, des expériences complémentaires montrent que l'intérêt de notre contribution est plus marqué lorsque le volume de données d'entraînement est lui-même plus conséquent. Ces mêmes expériences indiquent que les jeux de données DEFT Rich ERE sont suffisamment proches des jeux de données TAC 2015 et sont donc tout indiqués pour entraîner des modèles s'évaluant sur ce corpus.

Notre modèle global étant fortement tributaire de la qualité des étapes de détection des mentions d'entités et de leurs liens, une première piste d'extension évidente consiste à améliorer ces traitements. En effet, pour des soucis de temps et de compatibilité entre ces deux étapes, nous n'avons considéré ici que la détection d'entités nommées et la résolution de coréférences par l'outil Stanford CoreNLP et l'influence des paramètres de ces deux étapes n'a pas été analysée. Une étude comparative des performances intrinsèques de l'étape de résolution de coréférences entre les éditions 2015 et 2017 permettrait notamment de mieux appréhender l'impact de ce processus. Une configuration plus fine de ces traitements pourrait ainsi permettre une amélioration de la pertinence des contextes extraits et donc des représentations extraites. L'utilisation de modèles de désambiguïsation d'entités, en sus du modèle de coréférence, permettrait également de renforcer la qualité des liens entre mentions d'entités, voire de considérer un contexte inter-document. Dans ce cas, il serait également nécessaire de concevoir une heuristique de filtrage afin de ne pas considérer des contextes trop volumineux. Dans la continuité de cette idée, une deuxième piste d'extension serait de substituer le max-pooling utilisé pour agréger les différentes mentions d'entités par un mécanisme d'attention qui permettrait d'apprendre

à discriminer et filtrer les phrases de contexte selon leur pertinence. Cette idée serait donc complémentaire du filtrage préalable, toutes deux permettant de restreindre plus finement les informations du contexte à prendre en compte.

Les pistes évoquées jusqu'ici concernent l'amélioration de la prise en compte du contexte inter-phrastique. Il est également envisageable d'améliorer le traitement du contexte intra-phrastique par le modèle local en exploitant plus finement les dépendances syntaxiques structurant la convolution de graphe. Pour ce faire, il pourrait être intéressant de pouvoir traiter distinctement les différents types de dépendances syntaxiques. Afin de ne pas augmenter linéairement le nombre de paramètres du modèle avec le nombre de relations possibles dans le graphe, nous utiliserions alors la décomposition en bases (au sens des espaces vectoriels) des matrices de poids proposée par Schlichtkrull *et al.* (2018) : un nombre restreint de matrices-bases sont définies et communes à l'ensemble des dépendances. Pour chaque type de dépendance, des coefficients associés à chaque base sont modifiés pendant l'apprentissage et la matrice correspondant à une dépendance est obtenue par la moyenne pondérée des bases. Seuls les coefficients étant spécifiques à chaque dépendance, cette approche permet ainsi de combiner spécificité et compacité de la convolution de graphe. De plus, nous n'avons considéré dans ces travaux que les *Universal Dependencies* basiques. L'utilisation des dépendances améliorées pourrait également s'avérer intéressante puisque celles-ci sont plus compactes, raccourcissant de fait la distance entre les mots d'intérêt (tels que les entités) et le mot cible.

Chapitre 5

Extraction globale d'événement par Apprentissage Statistique Relationnel

Sommaire

Comme nous l'avons déjà vu dans les précédents chapitres, la résolution de la tâche d'extraction d'événements implique une prise en compte fine du contexte à la fois intra-phrastique et inter-phrastique. Nous avons jusqu'à présent pris en compte ce contexte inter-phrastique par le biais d'une représentation vectorielle de contexte. Si une représentation générique du contexte inter-phrastique améliore les prédictions d'un modèle neuronal local, nous avons montré dans le chapitre 3 l'intérêt d'utiliser une représentation spécifique à la tâche. Toutefois, cette représentation est statique et commune à l'ensemble du document. Nous avons donc étudié dans le chapitre 4 l'intérêt de produire une représentation du contexte inter-phrastique non seulement spécifique à la tâche mais également à la cible de prédiction. La méthode que nous avons proposée, utilisant les coréférences entre mentions d'entités pour identifier les phrases les plus pertinentes du contexte inter-phrastique, a également permis d'obtenir un gain important. La continuité logique de ces travaux serait donc d'affiner plus encore la prise en compte des interactions entre les différents événements à l'échelle du document. Ces deux précédents modèles fonctionnent selon une modélisation comparable : au niveau local, ou intra-phrastique, un modèle neuronal exploite finement les attributs sémantiques et syntaxiques des mots et leurs interactions au sein de la phrase. Au niveau inter-phrastique, une représentation du contexte est produite puis intégrée au modèle local pour enrichir la représentation exploitée. Le contexte inter-phrastique est donc constitué d'un seul vecteur tandis que chaque mot de la phrase contenant le mot cible possède sa représentation spécifique. Cette approche traite ainsi les informations locales et globales à des granularités différentes. Cette modélisation par-

cimonieuse permet ainsi de limiter la quantité et la complexité des données à traiter par le modèle tout en lui permettant de tirer profit d'informations distantes pouvant enrichir et clarifier des ambiguïtés au sein du contexte local. Cependant, ces modèles reposent sur l'hypothèse implicite que chaque ambiguïté locale est marginale et peut être résolue indépendamment des autres et séquentiellement en exploitant la présence d'informations plus explicites au sein du contexte. Cette hypothèse optimiste n'est cependant pas toujours vraie et certaines ambiguïtés gagnent à être envisagées et résolues conjointement.

Cette approche, dite de prédiction globale, vise à déterminer l'ensemble des prédictions des événements du document de manière à maximiser la cohérence globale au niveau du document. Les modèles neuronaux ne sont toutefois pas adaptés à ce type de modélisation. Nous l'expliquons plus en détail dans la première section de ce chapitre, où nous présentons également les approches d'apprentissage statistique relationnel (*Statistical Relational Learning* ou SRL) comme solution permettant une telle modélisation. Ces approches visent justement à définir un ensemble de contraintes globales entre différentes instances de prédictions et à optimiser la satisfaction globale de ces contraintes.

Nous détaillons plus particulièrement le langage de programmation probabiliste *Probabilistic Soft Learning* (ou PSL) que nous employons par la suite, permettant de définir des modèles graphiques probabilistes sur lesquels réaliser des inférences globales.

Le reste du chapitre présente deux études exploratoires quant à l'intérêt de cette modélisation. Ces expériences reposent sur l'utilisation des prédictions d'un modèle local de l'état de l'art pour définir des a priori locaux sur la présence d'événements qui seront optimisés globalement par le modèle PSL. Nous cherchons dans un premier temps à réaliser une prédiction globale des déclencheurs événementiels à l'échelle du document. La modélisation proposée permet d'obtenir un gain significatif sur le jeu de validation mais un gain plus faible et non-significatif en test.

Dans un second temps, nous nous intéressons à l'extraction des arguments. Afin d'isoler cette tâche des performances de la détection de déclencheurs, nous développons cette approche en considérant la détection d'événements comme préalablement appliquée. Nous soulevons alors une ambiguïté concernant les conditions d'évaluation utilisées par le mo-

dèle local. Nous nous plaçons ensuite dans ce que nous supposons être les mêmes conditions et notre modèle PSL obtient de très importants gains de performances. Nous montrons cependant dans une dernière section d'analyse qu'en s'évaluant dans des conditions plus rigoureuses, notre modèle obtient un gain plus ténu. Enfin, nous analysons plus précisément l'influence des différentes règles et concluons que l'implémentation actuelle de PSL ne permet pas de correctement modéliser nos contraintes.

5.1 Prédiction globale, limites des modèles neuronaux

L'objectif de ce chapitre est de proposer un modèle permettant de réaliser une classification jointe et simultanée de l'ensemble des événements d'un document en exploitant les interdépendances entre événements, arguments et entités. Réaliser une telle prédiction revient à maximiser la cohérence globale des prédictions. Les approches neuronales existantes ne permettent pas une telle optimisation. Dans le cas des approches convolutives existantes, le max pooling fait disparaître la dimension temporelle et impose l'usage d'un plongement de position centré sur la cible. Ainsi, il n'est possible de réaliser la prédiction que séquentiellement, un mot après l'autre. Concernant les approches récurrentes et celles utilisant la convolution de graphe (GCN), on dispose bien d'une représentation pour chaque mot de la phrase. Il est donc possible d'optimiser globalement cette séquence (ou graphe pour les GCNs) d'entrée en y adjoignant une couche de champ aléatoire conditionnel (*Conditional Random Field* ou CRF, Lafferty *et al.* (2001)). Cependant, bien que cette modélisation soit utilisable à l'échelle de la phrase, elle n'est pas envisageable à l'échelle d'un document compte tenu du fait que la complexité du modèle augmente exponentiellement avec la longueur des séquences considérées.

Il est donc nécessaire de se tourner vers d'autres formalismes à même d'exploiter à la fois la structure relationnelle d'un document et l'incertain inhérent à la nature non structurée du texte. L'apprentissage automatique relationnel est un sous-domaine de l'apprentissage automatique répondant spécifiquement à ce besoin. Nous le présentons de manière générale dans la prochaine sous-section. Parmi les nombreux formalismes SRL, nous nous intéressons au langage PSL, que nous décrivons par la suite.

5.1.1 Apprentissage automatique relationnel

L'apprentissage automatique relationnel (*Statistical Relational Learning* (SRL)) est un sous-domaine de l'apprentissage automatique s'intéressant à la modélisation de domaines présentant des structures relationnelles complexes et d'informations incertaines. Il se distingue donc des modèles d'apprentissage statistique par la nature des données exploitées. Les modèles d'apprentissage statistique opèrent sur une représentation vectorielle d'un objet, un mot cible dans le cadre de l'EE. Pour des approches non neuronales, cette représentation est constituée d'un ensemble de traits sélectionnés tandis que les approches neuronales produisent des représentations à l'aide d'une étape d'extraction de descripteurs, que celle-ci repose sur un CNN, un RNN ou un GCN. Les transformations de la représentation d'entrée opérées par l'extracteur de descripteurs, notamment dans le cas du modèle GCN, peuvent ainsi exploiter la structure relationnelle de la phrase. Cependant, lors de l'application du classifieur, approches neuronales comme non neuronales considèrent chaque objet indépendamment des autres. Seule sa représentation est prise en compte pour déterminer la prédiction de sortie. Formellement, ces modèles opèrent donc sur des données propositionnelles, constituées d'objets de même nature (des mots) et donc tous caractérisés par les mêmes attributs. De ce fait, si une partie de l'information de structure peut être exprimée à travers les attributs, c'est dans une forme fortement implicite et condensée. À l'inverse, les modèles SRL opèrent sur des données relationnelles : la représentation d'un objet en SRL est notamment caractérisée par sa relation avec d'autres objets qui peuvent être de nature différente, tels que des phrases ou des entités. Les modèles SRL ont ainsi pour objectif de modéliser une distribution jointe sur ces données relationnelles. Pour illustrer ceci, nous pouvons définir une contrainte exprimant que les types d'événements *Die* et *Attack* apparaissent ensemble dans un document. Considérons maintenant un document contenant un candidat présentant une ambiguïté entre les types *End-Position* et *Attack*, où l'on tendrait à assigner *End-Position* et un candidat pour lequel l'ambiguïté est entre la classe NULLE et la classe *Die*, où l'on tendrait à assigner *Die*. En considérant le contexte inter-phrastique, un modèle neuronal comme celui présenté dans le chapitre 3, traitant séquentiellement les exemples, tendrait à corriger

le premier déclencheur en *Attack*, compte tenu de la présence d'un événement de type *Die*, puis à corriger le deuxième exemple en NULLE. À l'inverse une optimisation globale permettrait de produire les couples [*Attack*/*Die*] ou [*End-Position*/*NULLE*] reflétant mieux la cohérence globale de la contrainte exprimée. Plusieurs des principaux paradigmes de SRL, tels que MLN (*Markov Logic Network* (Richardson et Domingos, 2006)) et PSL (*Probabilistic Soft Logic* (Richardson et Domingos, 2006)), ont recours à la logique du premier ordre pour exprimer des contraintes spécifiant un modèle graphique probabiliste qui est ensuite optimisé. Toutefois, comme nous le présentons dans la section suivante, la logique du premier ordre ne permet pas d'exprimer l'incertain et ces paradigmes reposent donc sur des redéfinitions assouplies de cette logique.

5.1.2 Limites de la logique du premier ordre

Une base de connaissances du premier ordre est constituée d'un ensemble de formules en logique du premier ordre (Genesereth et Nilsson, 1987). Ces formules sont construites à l'aide de 4 types de symboles : des constantes, des variables, des fonctions et des prédicats. Afin d'illustrer ce vocabulaire avec un exemple simple, nous nous plaçons ici dans le cadre de la modélisation d'un réseau social où l'on cherche à inférer la consommation de tabac.

Les constantes représentent des objets du domaine modélisé (p.ex. des personnes spécifiques, Anna, Bob). Les variables sont des symboles pouvant prendre des valeurs parmi les constantes (p.ex. x). Les fonctions associent un tuple d'objets à un objet (p.ex. VoisinDe). Les prédicats peuvent représenter des attributs d'objets (p.ex Fume) ou des relations entre objets (p.ex. Amis). Un terme est un symbole représentant un objet, c'est-à-dire soit une constante, soit une variable, soit l'application d'une fonction à un tuple de termes (p.ex. VoisinDe(Bob)). L'application d'un prédicat à un tuple de termes est appelée atome (ou formule atomique). Un atome est un littéral positif et la négation d'un atome, un littéral négatif.

Il est possible de construire récursivement des formules complexes par combinaison d'autres formules à l'aide de quantificateurs ($\exists$, $\forall$) et de connecteurs logiques ($\neg$, $\vee$, $\wedge$, $\Rightarrow$, $\Leftarrow$, $\Leftrightarrow$). Le fait que des amis tendent à avoir la même consommation (ou non

consommation) de tabac peut par exemple être exprimé ainsi :

$$\forall x \forall y \text{Amis}(x, y) \Rightarrow (\text{Fume}(x) \Leftrightarrow \text{Fume}(y)) \tag{5.1}$$

où $Amis(x, y)$, $Fume(x)$ et $Fume(y)$ sont tous des atomes (donc des littéraux positifs).

Un terme ne contenant aucune variable est appelé terme clos. De même, un prédicat clos est un prédicat ne faisant intervenir que des termes clos. L'ancrage d'une formule consiste à produire une instance de cette formule où chaque prédicat est clos. On appelle interprétation, ou monde possible, l'assignation d'une valeur de vérité à l'ensemble des atomes clos possibles. Dans notre exemple, cela reviendrait à déterminer les liens d'amitiés et la consommation de tabac de l'ensemble des personnes. Bien que la formule présentée soit vraie dans la majorité des cas, il existera nécessairement des contre-exemples. De fait, la restriction à la logique du premier ordre limiterait essentiellement la base à des formules triviales peu utiles pour des applications réelles. C'est pourquoi, malgré l'expressivité permise par ce formalisme, il est nécessaire d'assouplir ces contraintes, comme le proposent les différents paradigmes de SRL. La violation d'une formule n'y invalide plus complètement un monde mais réduit sa probabilité. En l'absence de monde parfaitement satisfaisant, le monde le plus probable est alors celui satisfaisant le plus de formules. Afin de pouvoir quantifier l'importance relative des différentes formules, on associe à chacune un poids. Plus le poids est élevé, plus la différence de probabilité entre un monde satisfaisant une formule et un autre ne la satisfaisant pas est importante.

5.2 Probabilistic Soft Logic

Parmi les paradigmes reposant sur cet assouplissement de la logique du premier ordre afin de modéliser l'incertain, les plus communs, notamment en extraction d'information (Pawar *et al.*, 2018; Rospocher, 2018), sont MLN et PSL.

À la différence de PSL, MLN ne permet pas de modéliser le partitionnement des données en documents et phrases. Comme nous l'expliquons plus loin dans cette section, ce mécanisme, disponible en PSL, est nécessaire pour éviter de considérer les instanciations

de formules impossibles, tel un événement dont le déclencheur et les arguments appartiennent à différents documents. Au regard de la combinatoire élevée qu'impose cette absence de partitionnement, les modèles MLN ne sont pas adaptés. De plus, nous nous inspirons dans ce chapitre de la modélisation proposée par Liu *et al.* (2016b) reposant également sur PSL. Nous avons donc choisi ce formalisme pour le reste de ce chapitre. Formellement, PSL est un langage de programmation probabiliste permettant de définir des champs aléatoires de Markov à marge maximale (hinge loss Markov Random Field où HL-MRF) à l'aide de règles exprimées en logique du premier ordre. Nous présentons dans une première sous-section l'extension de la logique du premier ordre utilisé par PSL puis le modèle HL-MRF dans une deuxième sous-section. Enfin, nous décrivons la procédure d'optimisation visant à déterminer l'assignation optimale des variables du graphe.

5.2.1 Logique de Łukasiewicz

À la différence de la logique du premier ordre présentée précédemment, PSL opère sur des variables réelles. Ceci permet de modéliser des informations naturellement continues, telles que la similarité ou des concepts vagues. Les valeurs de vérité des variables peuvent alors prendre n'importe quelle valeur de l'intervalle $[0, 1]$. Cependant, l'extension des variables au domaine réel impose une extension similaire des opérateurs booléens, ne pouvant traiter dans leur définition classique que des valeurs binaires. PSL utilise donc les opérateurs de la logique de Łukasiewicz, spécifiquement adaptés à ce cas de figure. Pour ce faire, les opérateurs booléens sont redéfinis ainsi :

$$x \wedge y = max(x + y - 1, 0) \tag{5.2}$$

$$x \vee y = min(x + y, 1) \tag{5.3}$$

$$\neg x = 1 - x \tag{5.4}$$

Les opérateurs de conjonction et de disjonction (respectivement eq. 5.2 et 5.3) sont respectivement les t-norme et t-co-norme de Łukasiewicz. Comme nous le détaillons plus avant dans la section suivante, notre modèle PSL exploite les prédictions d'un modèle

neuronal local en tant qu'a priori pour le modèle global. Les prédictions du modèle local n'étant pas binaires, nous pourrons ainsi directement utiliser les valeurs associées aux différentes classes. Ceci peut permettre de distinguer les prédictions fiables des cas ambigus où plusieurs classes se voient associer des probabilités proches.

5.2.2 Satisfiabilité des règles

Le langage PSL consiste à définir un ensemble de règles[1] qui peuvent ou non être associées à des poids. Les règles sans poids sont appelées contraintes fortes car un monde ne satisfaisant pas ces règles est invalide. Dans ce cas, la règle se termine par un point. À l'inverse, les règles associées à des poids sont appelées contraintes faibles car un monde dans lequel elles ne sont pas satisfaites est seulement moins probable. PSL permet de définir deux types de règles, des règles logiques et des règles arithmétiques.

Règles logiques

Une règle logique est une implication logique dont la partie gauche est appelée "Corps" et doit être une conjonction de prédicats et la partie droite, "Tête" est constituée d'une disjonction de prédicats.

$$\text{Amis}(x, y) \wedge \text{Fume}(x) \Rightarrow \text{Fume}(y) \tag{5.5}$$

Ainsi, dans la règle 5.5, le corps est $\text{Amis}(x, y) \wedge \text{Fume}(x)$ et la tête $\text{Fume}(y)$.

Pour qu'une règle soit considérée comme satisfaite, la valeur de vérité de la tête doit être supérieure ou égale à celle du corps. Comme pour une implication en logique du premier ordre, lorsque les valeurs de vérité des prédicats du corps ont des valeurs de vérité faible, la règle sera satisfaite indépendamment de la valeur de vérité de la tête. Par

1. Le terme règle, spécifique à PSL, englobe à la fois les règles logiques, équivalentes aux formules définies précédemment, et les règles arithmétiques.

exemple, avec $\text{Amis}(x, y) = 0.2$, $\text{Fume}(x) = 0.3$ [2] :

$$\text{Amis}(x, y) \wedge \text{Fume}(x) = max(0, 0.3 + 0.2 - 1) \qquad = 0$$

Dans le cas contraire, par exemple avec $\text{Amis}(x, y) = 0.7$ et $\text{Fume}(x) = 0.8$, la valeur de vérité de $\text{Fume}(y)$ doit être supérieure à 0.5 pour que la règle soit satisfaite.

$$\text{Amis}(x, y) \wedge \text{Fume}(x) = max(0, 0.7 + 0.8 - 1) \qquad = 0.5$$

L'objectif du modèle PSL étant la maximisation de la satisfiabilité des règles, on ne s'intéresse pas seulement à l'évaluation binaire de la satisfaction de la règle (i.e. la règle est satisfaite ou non) mais surtout à sa distance de satisfaction.

En remplaçant les opérateurs par leur forme de Łukasiewicz on peut déduire de la forme 5.6 la forme 5.7 de l'expression de la satisfaction d'une règle générique :

$$B_1(X) \wedge \ldots \wedge B_n(X) \Rightarrow H_1(X) \vee \ldots \vee H_n(X) \tag{5.6}$$

$$\begin{aligned}
&= \neg\big(B_1(X) \wedge \ldots \wedge B_n(X)\big) \vee H_1(X) \vee \ldots \vee H_n(x) \\
&= \neg\big(B_1(X) \wedge \ldots \wedge B_n(X) \wedge \neg(H_1(X) \vee \ldots \vee H_n(x))\big) \\
&= 1 - max\Big(\wedge \big(B_1(X), \ldots, B_n(X)\big) + \neg \vee \big(H_1(X), \ldots, H_n(x)\big) - 1, 0 \Big) \\
&= 1 - max\Big(\wedge \big(B_1(X), \ldots, B_n(X)\big) + 1 - \vee\big(H_1(X), \ldots, H_n(x)\big) - 1, 0 \Big) \\
&= 1 - max\Big(\wedge \big(B_1(X), \ldots, B_n(X)\big) - \vee\big(H_1(X), \ldots, H_n(x)\big), 0 \Big)
\end{aligned} \tag{5.7}$$

En faisant la différence entre la satisfiabilité maximale, 1, et la forme 5.7 de la satisfaction effective de la règle, on obtient :

$$max\Big(\wedge \big(B_1(X), \ldots, B_n(X)\big) - \vee\big(H_1(X), \ldots, H_n(x)\big), 0 \Big) \tag{5.8}$$

2. Afin d'éviter les ambiguïtés entre la virgule de séparation décimale et celle séparant les paramètres des fonctions min et max utilisées, nous employons dans cette section le point comme séparateur décimal.

Formellement, l'équation 5.8 exprime sous forme de *hinge-loss* la distance de satisfaction $d_r(I)$ d'une règle r pour une interprétation I.

En instanciant l'exemple 5.5 avec les valeurs $\text{Amis}(x, y) = 0.7$, $\text{Fume}(x) = 0.8$ et $\text{Fume}(y) = 0.2$, nous obtenons ainsi une distance de satisfaction :

$$\begin{aligned} & max\Big(\wedge \big(Amis(x,y), Fume(x)\big) - Fume(y), 0 \Big) \\ & = max\Big(max(0.7 + 0.8 - 1,\ 0) - min(0,2,\ 1), 0 \Big) \\ & = max\Big(0.5 - 0.2, 0 \Big) \\ & = 0.3 \end{aligned}$$

Règles arithmétiques

Les règles arithmétiques permettent d'exprimer des égalités ou inégalités entre atomes. Elles peuvent également utiliser des atomes de sommation pour exprimer la somme de tous les atomes clos possibles par substitution de variables de sommation. Pour les marquer, ces variables sont préfixées du symbole "+".

$$\text{Amis}(x, +y) > 1. \tag{5.9}$$

Par exemple, la contrainte forte 5.9 indique que la somme des valeurs des relations d'amitié d'un individu est supérieure à 1. Il est également possible d'appliquer un filtre à la variable de sommation. Le filtre consiste en un prédicat associé à la variable. Ainsi, pour indiquer cette fois que la somme des relations d'amitié d'un individu avec des individus fumeurs est supérieure à 1, la contrainte 5.9 est transformée en :

$$\text{Amis}(x, +y) > 1.\{y : \text{Fume}(y)\} \tag{5.10}$$

La production d'un modèle HL-MRF nécessite de calculer la distance de satisfaction de l'ensemble des règles ancrées. Le nombre d'ancrages possibles augmentant exponentiellement avec le nombre d'atomes clos possibles, il peut être nécessaire de limiter ce

nombre d'ancrages. À cette fin, PSL fournit un mécanisme de blocs : les blocs sont des prédicats binaires spécifiques permettant de circonscrire l'ancrage d'une règle à un nombre restreint de combinaisons de variables. Dans l'exemple de la consommation de tabac, nous pourrions souhaiter modéliser le lien entre amitié et consommation de tabac en prenant en compte les spécificités de chaque ville. Si nous souhaitions ne considérer l'influence de l'amitié sur la consommation de tabac que pour des gens d'une même ville, nous pourrions introduire un prédicat HabiteA pour compléter la règle 5.5 ainsi :

$$\text{HabiteA}(x, z) \wedge \text{HabiteA}(y, z) \wedge \text{Amis}(x, y) \wedge \text{Fume}(x) \Rightarrow \text{Fume}(y) \tag{5.11}$$

Avec l'utilisation d'un prédicat classique, cette règle sera sans effet lorsque les deux personnes considérées habitent dans des villes différentes puisque le corps de la règle sera faux et donc celle-ci trivialement satisfaite. Cependant, l'ancrage sera toujours réalisé pour l'ensemble des combinaisons de personnes possibles. À l'inverse, en définissant HabiteA en tant que prédicat bloc, la validité de ces atomes sera évaluée en premier et seules les instanciations valides seront produites. La distinction entre les prédicats classiques et blocs n'étant pas visible dans la définition des règles, nous les indiquerons en gras dans les prochaines sections.

5.2.3 Apprentissage et inférence

La définition d'un modèle HL-MRF consiste à produire une fonction de densité d'une interprétation I. Pour un ensemble R de règles chacune associées à un poids λ_r, PSL produit l'ensemble des ancrages possibles des règles avec l'interprétation I afin d'obtenir sa fonction de densité $f(I)$:

$$f(I) = \frac{1}{Z} exp\big[- \sum_{r \in R} \lambda_r (d_r(I))^{p_r} \big] \tag{5.12}$$

Dans les faits, nous cherchons à utiliser ce modèle pour inférer l'assignation d'une partie Y des prédicats de l'interprétation, les cibles, à partir d'une partie X, les observations, dont les valeurs de vérité sont connues. Formellement, il s'agit de déterminer l'assignation

la plus probable en maximisant la probabilité conditionnelle des cibles en fonction des observations.

$$\underset{Y}{\operatorname{argmax}}\, P(Y|X) = \frac{1}{Z} exp\Big[- \sum_{r \in R} \lambda_r (d_r(X,Y))^{p_r} \Big] \tag{5.13}$$

Pour obtenir un modèle HL-MRF apte à prédire de nouvelles données, il est nécessaire de déterminer les poids optimaux des règles. Pour réaliser cet apprentissage, PSL utilise un perceptron structuré pour déterminer les poids maximisant le maximum de vraisemblance des données d'apprentissage. Le nombre de paramètres (i.e. le nombre de règles) étant relativement restreint, il est également possible de déterminer empiriquement ces poids à l'aide d'une recherche d'hyperparamètres. Le langage PSL étant assez récent, la procédure d'apprentissage n'est pas mature et des expériences préliminaires n'ont pas permis d'obtenir des poids satisfaisants. Nous utilisons donc la seconde solution pour le reste de ces travaux.

5.3 Prédiction globale des déclencheurs événementiels

Que ce soit au niveau inter-phrastique comme nous l'avons déjà exploré dans les précédents chapitres ou au niveau intra-phrastique, il existe une forte interdépendance entre les occurrences des différents événements au sein d'un document. Il est également possible de tirer profit des interdépendances entre mentions d'entités et arguments ; mais nous nous focalisons dans un premier temps sur les déclencheurs. Nous cherchons donc ici à exploiter la probabilité conditionnelle d'observer une mention d'un type d'événement au sein d'une phrase ou d'un document sachant la présence d'une mention d'un autre événement. Ce type d'interdépendance peut être formalisé sous forme de règles PSL pour réaliser une optimisation globale des prédictions. Le nombre d'ancrages à évaluer augmentant de manière exponentielle avec le nombre de candidats, il n'est pas envisageable d'appliquer notre modèle à l'ensemble des mots du document. De plus, la représentation sémantique du candidat et la modélisation fine du contexte local, telle que permise respectivement par des plongements et des modèles neuronaux, seraient difficilement transposables sous forme de règles PSL. C'est pourquoi, comme Liu *et al.* (2016b) dont nous nous inspirons pour

cette section, nous appliquons préalablement un modèle statistique local aux documents. Chaque mot pour lequel la probabilité de la classe NULLE est inférieure à un seuil p est conservé comme candidat. La valeur de ce seuil, $p = 0,9$, a été déterminée empiriquement lors d'analyses préliminaires des données d'entraînement pour permettre de réduire drastiquement le nombre de candidats à considérer tout en conservant la possibilité de corriger les faux négatifs du modèle local.

Nous tentons en premier lieu de reproduire une partie des règles proposées par Liu *et al.* (2016b) et de les adapter à notre modèle local, différent de celui qu'ils utilisent. Contrairement à ces travaux, nous ne considérons pas l'utilisation de l'Allocation de Dirichlet Latente (LDA) (Blei *et al.*, 2003) pour modéliser les thématiques des documents. Les règles que nous produisons concernent donc les probabilités conditionnelles entre types d'événements à l'échelle du document ou de la phrase.

5.3.1 Définition du modèle

Afin de définir un modèle PSL, il est nécessaire d'introduire un certain nombre de variables, prédicats et règles que nous détaillons ici.

Variables

Bien que PSL n'impose pas de déclarer préalablement de types de variables, nous nous astreignons à la convention suivante, les variables étant suffixées dans certains prédicats d'un nombre pour distinguer deux variables du même type (ex : *t1*, *t2*). Nous utilisons les variables :

— ***w*** : identifiant unique d'un déclencheur/candidat ;

— ***t*** : un des types d'événements ou la classe "NULLE" ;

— ***s*** : identifiant unique d'une phrase ;

— ***d*** : identifiant unique d'un document.

Prédicats

On distingue deux types de prédicats. Les prédicats fermés, ou observables, sont ceux pour lesquels on fournit une liste exhaustive des instanciations vraies, les autres instanciations étant considérées comme fausses par défaut. Les prédicats blocs sont un type spécifique de prédicats fermés. À l'inverse, les prédicats ouverts, ou cibles, sont ceux auxquels l'inférence doit assigner une valeur de vérité.

Notre modèle PSL est ainsi défini à l'aide des prédicats observables suivants :

— **candEvt(*w*, *t*)** : probabilité de la classe t attribuée au candidat w par le modèle local ;

— **sentLevel(*t1*, *t2*)** : probabilité d'observer un événement de type t2 sachant la présence d'un événement de type t1 dans une phrase ;

— **docLevel(*t1*, *t2*)** : probabilité d'observer un événement de type t2 sachant la présence d'un événement de type t1 dans un document ;

ainsi que des prédicats blocs de phrase et de document :

— **inSent(*w*, *s*)** : w appartient à la phrase s ;

— **inDoc(*w*, *d*)** : w appartient au document d ;

et du prédicat cible suivant :

— **isTriggerType(*w*, *t*)** : le candidat w est déclencheur d'un événement de type t.

Règles

À partir des prédicats définis précédemment, nous introduisons les règles suivantes :

1. **contrainte forte :** la somme des probabilités associées aux différentes classes est égale à 1 pour chaque candidat :

$$\text{isTriggerType}(w, +t) = 1. \tag{5.14}$$

2. les candidats sont généralement des déclencheurs et non des faux positifs :

$$\neg\text{isTriggerType}(w, \text{'NULLE'})$$

3. les prédictions du modèle local sont correctes :

$$\text{candEvt(w,t)} \Rightarrow \text{isTriggerType(w,t)} \tag{5.15}$$

4. si un déclencheur d'un événement est présent dans une phrase et que l'association phrastique est forte avec le type d'événement d'un candidat présent dans la phrase, le candidat est bien un déclencheur du type concerné :

$$\begin{aligned}&\textbf{inSent}(w1, s) \ \wedge \ \textbf{inSent}(w2, s) \ \wedge \ \text{sentLevel}(t1, t2) \\ &\quad \wedge \ \text{candEvt}(w2, t2) \ \wedge \ \text{isTriggerType}(w1, t1) \Rightarrow \text{isTriggerType}(w2, t2)\end{aligned} \tag{5.16}$$

5. si un déclencheur d'un événement est présent dans un document et que l'association au niveau document est forte avec le type d'événement d'un candidat présent dans le document, le candidat est bien un déclencheur du type concerné :

$$\begin{aligned}&\textbf{inDoc}(w1, d) \ \wedge \ \textbf{inDoc}(w2, d) \ \wedge \ \text{docLevel}(t1, t2) \\ &\quad \wedge \ \text{candEvt}(w2, t2) \ \wedge \ \text{isTriggerType}(w1, t1) \Rightarrow \text{isTriggerType}(w2, t2)\end{aligned} \tag{5.17}$$

5.3.2 Production des données

Afin de procéder à l'inférence du modèle, il est nécessaire de produire la liste exhaustive des observations des prédicats fermés (candEvt, sentLevel, docLevel, inSent, inDoc) et la liste des cibles de prédictions pour le prédicat isTriggerType.

Association événement-événement

Concernant les deux prédicats de dépendance entre événements, sentLevel et docLevel, nous produisons les statistiques sur le jeu d'apprentissage de la manière suivante. Pour chaque paire de types d'événements, $num_{doc}(t1, t2)$ et $num_{sent}(t1, t2)$ désignent respectivement le nombre de cooccurrences de ces deux événements dans un même document et une même phrase du jeu d'apprentissage. Nous obtenons alors les probabilités conditionnelles

$p_{evt-evt-[doc/sent]}(t2 \mid t1)$ à l'aide des formules suivantes :

$$p_{evt\text{-}evt\text{-}sent}(t2 \mid t1) = \frac{num_{sent}(t1, t2)}{\sum_{t \in T} num_{sent}(t, t1)}$$

$$p_{evt\text{-}evt\text{-}doc}(t2 \mid t1) = \frac{num_{doc}(t1, t2)}{\sum_{t \in T} num_{doc}(t, t1)}$$

Les probabilités obtenues sont alors utilisées pour définir les valeurs de vérité des prédicats correspondants.

A priori locaux

Puisque nous visons à améliorer les performances du modèle local, il est nécessaire de produire pour chaque candidat les observations candEvt(w,t) et les cibles isTrigger-Type(w,t) associées à la classe prédite par le modèle local ainsi qu'à d'autres classes. Nous définissons ainsi un paramètre *topPrior* spécifiant le nombre de classes renvoyées en plus de la classe NULLE.

Avec topPrior = j, pour chaque candidat, on ajoute la classe NULLE et les j meilleures classes non nulles à la liste des cibles (candEvt). On ajoute également des observations du prédicat candEvt pour ces classes. Pour chacune de ces observations, les valeurs de vérité correspondent à la probabilité attribuée par le modèle local à cette classe.

5.4 Prédiction globale des déclencheurs événementiels : expériences

5.4.1 Paramètres et ressources

Puisque l'objectif de ce chapitre est la prédiction jointe et globale des déclencheurs et des arguments, nous devons disposer des prédictions locales pour ces deux tâches. N'ayant pas nous-même réalisé d'expériences sur la tâche d'extraction d'arguments, nous avons utilisé le seul modèle de l'état de l'art ayant traité ces deux tâches et dont le code était disponible. Il s'agit du modèle proposé dans (Nguyen *et al.*, 2016a). Ce modèle repose

sur une architecture récurrente et permet une prédiction jointe des déclencheurs et des arguments sur le jeu de données ACE. Le système développé utilise un format de données d'entrée spécifique et difficilement reproductible. Les auteurs nous ayant gracieusement fourni les données ACE prétraitées mais pas le script de prétraitement, il ne nous a pas été possible d'appliquer ce système aux données TAC. On peut donc regretter ici l'impossibilité de comparer cette étude aux autres contributions présentées dans les chapitres précédents. Nous réaliserons donc les expériences sur le jeu de données ACE, doté de 33 types d'événements, en utilisant la répartition usuelle entre jeu d'apprentissage, de validation et de test. Comme nous l'avons déjà évoqué, nous ne réalisons pas à proprement parler d'apprentissage sur les poids des règles mais une recherche empirique de la meilleure configuration. Pour ce faire, nous utilisons l'algorithme d'optimisation bayésienne d'hyperparamètres fourni par la librairie hyperopt. Nous fixons le poids de la règle directe (formule 5.15) à 8 et faisons varier le poids des autres règles de 0 à 15. Afin de ne considérer que l'influence de notre modélisation sur les performances, 10 entraînements du modèle local sont réalisés mais seules les prédictions du meilleur modèle sont considérées pour l'évaluation des performances et la production des données des modèles PSL. Les performances des modèles PSL sont toujours des moyennes de 10 inférences utilisant ces mêmes données.

5.4.2 Performances du modèle de base

Nous cherchons dans un premier temps à quantifier les performances du modèle précédemment défini et notamment l'influence du nombre d'a priori (topPrior) sur celles-ci. Concernant le choix du nombre d'a priori (topP), nous considérons la plage $[1-4]$. Nous présentons en Table 5.4.1 les performances du modèle local et de différentes configurations du modèle PSL sur le corpus de validation.

— modèle local : le modèle RNN de Nguyen *et al.* (2016a) dont les prédictions sont utilisées pour produire les observations des autres modèles ;

— PSL : performances de la meilleure configuration de poids des règles et de nombre d'a priori ;

TABLE 5.4.1 – Comparaison des performances en validation en fonction du nombre d'a priori considérés.

Configurations	**P**	**R**	**F**	**std**
modèle local	68,77	68,91	68,84	-
PSL	69,77	68,22	68,98	0,21
topPrior				
1	69,77	68,22	68,98	0,21
2	68,74	68,87	68,8	0,31
3	68,99	68,59	68,79	0,15
4	68,75	68,89	68,82	0,22

— topPrior[1/2/3/4] : pour chaque valeur du nombre d'a priori, nous présentons les performances de la meilleure configuration des poids des règles.

Le modèle local obtient un gain de seulement $0,14$, non significatif par rapport au modèle local (t-test à échantillon unique). De plus, les différentes configurations de poids des règles ne semblent pas influer sur la performance du modèle. Enfin, nous constatons que contrairement à notre hypothèse, l'augmentation du nombre d'a priori ne permet pas au modèle de corriger les prédictions du modèle local. Avec topPrior = 1, le modèle ne peut que corriger les faux négatifs et faux positifs mais pas les mauvaises prédictions entre deux classes d'événements. Ceci est à mettre en lien avec la prédominance de ces deux premiers types d'erreurs (respectivement 134 et 133 erreurs en validation) par rapport aux confusions entre types d'événements (24 erreurs).

5.4.3 Extension du modèle global

Puisque le modèle global semble essentiellement reproduire les prédictions du modèle local à travers la règle d'inférence directe (règle 5.15), nous proposons plusieurs variantes permettant de moduler ces prédictions afin d'améliorer les performances.

Corrélation par classe Les performances du modèle étant inégales entre les classes, il est souhaitable d'en tenir compte dans le modèle local. Le modèle pourra ainsi accorder une importance plus forte aux a priori des classes correctement prédites par le modèle local qu'aux classes généralement incorrectes. Nous utilisons la corrélation bisérielle afin d'obtenir un coefficient de corrélation entre les prédictions du modèle (variable quantitative) et

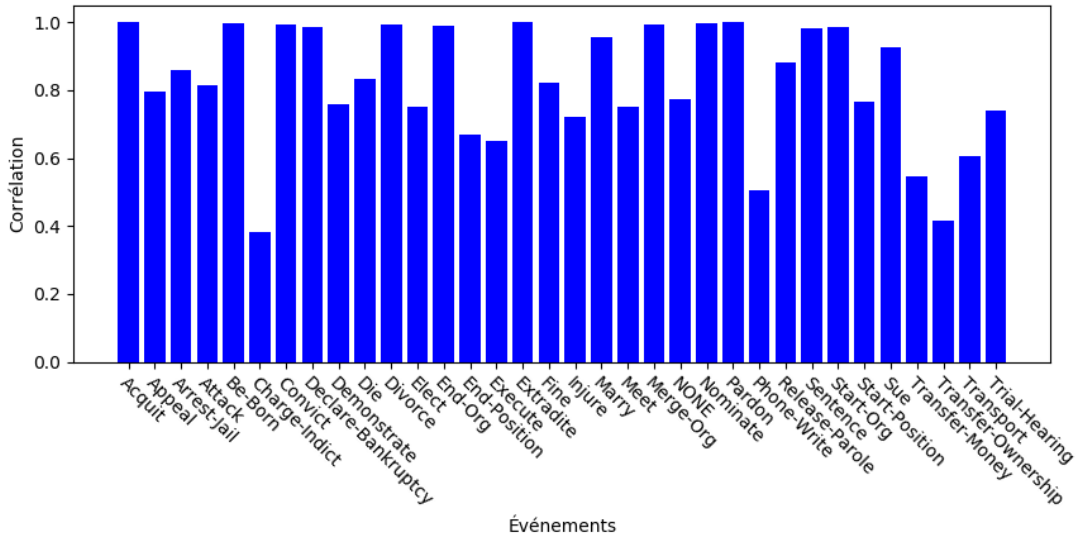

FIGURE 5.4.1 – Corrélation bisérielle par type d'événements.

les annotations (variable binaire). Ces corrélations sont produites sur le jeu de validation. En effet, calculer ces statistiques sur le jeu d'entraînement ne rendrait pas compte de la capacité de généralisation du modèle. La Figure 5.4.1 présente ces corrélations par classe. Afin d'introduire cette information dans notre modèle, nous supprimons la règle directe et la remplaçons par un ensemble de règles. Pour chaque type d'événement, nous créons une règle spécifique dont le poids est le produit de cette corrélation avec le poids de la règle directe. De plus, cette corrélation témoigne aussi de la capacité du modèle local à ne pas prédire une classe en l'absence de l'événement spécifique. Nous introduisons donc pour chaque type d'événement une règle "inverse", de même poids et définie comme suit :

$$\neg\text{candEvt(w,t)} \Rightarrow \neg\text{isTriggerType}(w, t)$$

Softmax avec température Afin de tirer profit des dépendances globales dont l'influence est jusqu'à présent négligeable face aux règles directes, nous modifions les prédictions locales utilisées pour générer les observations candEvt afin de réduire les écarts entre la classe dominante et les autres classes. Ces prédictions locales sont en effet obtenues par application d'un softmax aux logits du modèle, ce qui tend à amplifier la valeur du logit de la classe dominante tout en diminuant les valeurs des autres classes. Afin d'atténuer cet effet, nous appliquons un log à ces prédictions afin de retrouver les logits d'origine (à

TABLE 5.4.2 – Influence de la prise en compte de la corrélation bisérielle entre les prédictions du modèle local et les annotations (+Correl) et de l'utilisation du softmax avec température (+Température). Performances sur la base de test ACE 2005.

Configurations	**P**	**R**	**F**	**std**
Modèle local	65,37	68,64	66,96	-
PSL	67,14	67,3	67,22	0,3
+Correl	68,73	67,05	67,88	0,03
+Correl + Température	65,58	68,86	67,18	0

une constante près) avant d'appliquer un softmax avec température :

$$qi = \frac{exp(zi/T)}{\sum_j exp(zj/T)}$$

Une température $T = 1$ est équivalente à un softmax classique tandis qu'une température plus élevée rend la prédiction nuancée, réduisant l'écart entre les différentes classes.

Nous présentons, en Table 5.4.2, les résultats des performances du modèle avec ces extensions, sur le corpus de test. La modulation des règles directes et inverses en fonction du type d'événement permet d'obtenir un gain faible mais significatif par rapport au modèle PSL initial tandis que l'utilisation du softmax avec température ne permet pas d'obtenir un gain. Cependant, l'influence des poids des différentes règles (non présentées ici) semble négligeable. De plus, la variance des deux nouvelles modélisations est respectivement pratiquement nulle et nulle. En outre, les meilleures performances sont encore ici obtenues pour topPrior = 1. Il semble donc que le modèle ne prenne pas bien en compte les différentes interactions distantes modélisées et ne fasse encore que corriger marginalement les prédictions locales à l'aide des contraintes fortes.

5.5 Prédiction globale des arguments

En complément de la détection globale des déclencheurs, nous nous intéressons à l'extraction globale des arguments. Afin de mieux identifier les interactions utiles à exploiter pour cette tâche, nous considérons en premier lieu une situation idéale dans laquelle nous disposons de l'ensemble des déclencheurs événementiels correctement prédits.

TABLE 5.5.1 – Comparaison des performances sur la base de test ACE 2005 en fonction du type d'évaluation de la classe NULLE pour les arguments.

	Déclencheurs			Arguments		
Configurations	**P**	**R**	**F**	**P**	**R**	**F**
$\text{JRNN}_{\text{Officiel}}$	66.0	73.0	69.3	54.2	56.7	55.4
$\text{JRNN}_{\text{Repro}}$	65.37	68.64	66.96	55,85	57,46	56,64
$\text{JRNN}_{\text{Repro-Nulle}}$	65.37	68.64	66.96	51,14	24,53	33,16

5.5.1 Note sur les conditions d'évaluation des arguments

Comme pour les sections précédentes concernant les déclencheurs, nous cherchons ici à améliorer les prédictions d'arguments obtenues par le modèle de Nguyen *et al.* (2016a). Nous utilisons pour cela les données et le code mis à disposition par les auteurs. Cependant, puisque le programme d'évaluation n'était pas disponible, nous avons dû le ré-implémenter. Or, il ne nous a été possible d'obtenir des performances similaires à celles de l'article d'origine qu'en incluant la classe NULLE parmi les classes considérées pour le calcul des performances. Nous présentons en Table 5.5.1 les performances de l'article d'origine ($\text{JRNN}_{\text{Officiel}}$) ainsi que du modèle que nous avons entraîné à partir du code et des données d'origine en utilisant une évaluation normale ($\text{JRNN}_{\text{Repro}}$) et une évaluation incluant la classe NULLE ($\text{JRNN}_{\text{Repro-Nulle}}$).

Nous pouvons voir qu'entre les deux conditions d'évaluation de notre modèle reproduit, celles où la classe NULLE n'est pas considérée sont très inférieures, notamment au niveau du rappel. La classe négative étant majoritaire, ce phénomène s'explique par l'augmentation relative de la taille des autres classes, et donc de leur importance dans le calcul du rappel micro.

Or, la proximité des performances pour les déclencheurs entre la reproduction et les résultats de l'article semble indiquer que le code mis à disposition permet bien d'obtenir un modèle comparable à celui de l'article. Puisque les performances de ce modèle en détection d'arguments sont proches de celles de l'article lorsqu'on considère la classe NULLE, nous supposons que les auteurs s'évaluent dans ces conditions. Afin de pouvoir nous comparer à cette approche de l'état de l'art, nous avons fait le choix de nous évaluer dans les mêmes conditions que l'article dans les sections suivantes.

5.5.2 Définition du modèle

Variables

Nous utilisons ici logiquement les mêmes variables que celles précédemment utilisées pour les déclencheurs, complétées de celles nécessaires à la prise en compte des arguments :

— **w** : identifiant unique d'un déclencheur ;

— **t** : un des 33 types d'événement (+ classe NULLE) ;

— **em** : identifiant unique d'une mention d'entité ;

— **e** : identifiant unique d'une entité ;

— **r** : un des 35 types d'argument (+ classe NULLE) ;

— **s** : identifiant unique d'une phrase ;

— **d** : identifiant unique d'un document.

Puisque nous considérons ici acquise la prédiction des déclencheurs, la variable w ne désigne plus que des déclencheurs valides et non des candidats.

Prédicats

Nous définissons les prédicats suivants pour modéliser la tâche d'extraction d'arguments :

— **isTriggerType(*w*, *t*)** : w est déclencheur d'un événement de type t ;

— **candArg(*w*, *t*, *em*, *r*)** : probabilité assignée par le modèle local à la classification de la mention *em* en tant que rôle r dans l'événement de type t déclenché par w ;

— **sentLevelEventArg(*t*, *r*)** : probabilité de présence d'un argument de rôle r dans un document où apparaît un événement de type t ;

— **docLevelEventArg(*t*, *r*)** : probabilité de présence d'un argument de rôle r dans une phrase où apparaît un événement de type t ;

— **sentLevelArgArg(*r1*, *r2*)** : probabilité qu'une entité joue le rôle $r2$ sachant qu'elle tient également le rôle $r1$ dans cette phrase ;

— **docLevelArgArg(*r1*, *r2*)** : probabilité qu'une entité joue le rôle $r2$ sachant qu'elle tient également le rôle $r1$ dans ce document.

Afin d'exploiter les dépendances entre les mentions d'une même entité, nous définissons également le prédicat bloc suivant :

— **entityMention(*em*, *e*)** : la mention *em* fait référence à l'entité *e* ;

ainsi que les mêmes prédicats blocs de phrase et de document que pour la prédiction globale de déclencheurs. Le prédicat cible est :

— **isArg(*w*, *t*, *em*, *r*)** : *em* tient le rôle r dans l'événement de type t déclenché par w.

Règles

À partir des prédicats définis précédemment, nous introduisons les règles et contraintes fortes suivantes :

1. **contrainte forte :** une entité ne peut tenir un rôle que dans un événement détecté :

$$\neg\text{isTriggerType}(w,t) \Rightarrow \neg\text{isArg}(w,t,em,r).$$

2. **contrainte forte :** une entité n'a qu'un rôle par événement (classe NULLE incluse) :

$$\text{isArg}(w,t,em,+r) = 1.$$

3. le modèle local tend à prédire correctement les rôles :

$$\text{isTriggerType}(w,t) \wedge \text{candArg}(w,t,em,r) \Rightarrow isArg(w,t,em,r) \tag{5.18}$$

4. si un déclencheur d'un événement t est présent dans une phrase et que l'association phrastique est forte avec un rôle r prédit localement, la mention *em* est bien du

type r :

$$\text{isTriggerType(w1,t1)} \wedge \text{candArg(w2,t2,em2,r2)} \wedge \text{sentLevelEventArg(t1,r2)} \wedge \textbf{inSent(w1, s)} \wedge \textbf{inSent(em2, s)} \Rightarrow \text{isArg(w2,t2,em2,r2)}$$

5. si un déclencheur d'un événement est présent dans un document et que l'association au niveau document est forte avec le rôle d'un candidat au sein du même document, le candidat est bien un argument du type concerné :

$$\text{isTriggerType}(w1, t1) \wedge \text{candArg}(w2, t2, em2, r2) \wedge \text{docLevelEventArg}(t1, r2) \wedge \textbf{inDoc}(w1, d) \wedge \textbf{inDoc}(em2, d) \Rightarrow \text{isArg}(w2, t2, em2, r2)$$

6. si une mention d'entité $em1$ joue un rôle r_1 et qu'une autre mention, $em2$, fait partie d'une prédiction locale dont le rôle r_2 est fortement associé au r_1 au niveau du document alors le candidat est bien un argument :

$$\text{isArg}(w1, t1, em1, r1) \wedge \text{candArg}(w2, t2, em2, r2) \wedge \textbf{entityMention}(em1, e) \wedge \textbf{entityMention}(em2, e) \wedge \text{DocLevelArgArg}(r1, r2) \Rightarrow \text{isArg}(w2, t2, em2, r2)$$

7. si une mention $em1$ joue un r_1 et qu'elle fait partie d'un candidat dont le r_2 est fortement associé au r_1 au niveau phrastique alors le candidat est bien un argument :

$$\text{isArg}(w1, t1, em, r1) \wedge \text{candArg}(w2, t2, em, r2) \wedge \text{sentLevelArgArg}(r1, r2) \Rightarrow \text{isArg}(w2, t2, em, r2)$$

5.5.3 Production des données

Comme dans la précédente expérience, il est nécessaire de produire plusieurs statistiques pour constituer les observations de certains prédicats.

Certains événements étant très rares à l'échelle du jeu de données, leurs arguments

le sont d'autant plus, rendant les cooccurrences peu utiles. Pour calculer des statistiques événement-argument et argument-argument plus utiles, il peut donc être intéressant d'agréger certains types d'arguments communs à plusieurs types d'événements. L'argument *Adjudicator* par exemple est commun à l'ensemble des types d'événements juridiques et ces différentes versions sont sémantiquement proches. À l'inverse, l'argument *Agent* n'a pas de sens commun et sa signification est très différente d'un événement à un autre (*Start-Org_Agent* pour le fondateur d'une entreprise et *Execute_Agent* pour un bourreau). Les arguments temporels (*Time-After, Time-At-Beginning, Time-At-End, Time-Before Time-Ending, Time-Holds, Time-Starting, Time-Within*) étant très peu présents à l'exception de *Time-Within*, ils sont tous agrégés en ce dernier. Nous considérons plusieurs niveaux d'agrégation des types d'arguments :

— les 229 arguments **spécifiques** obtenus en distinguant les types d'événements ;

— les 35 arguments **génériques** ne distinguant pas le type d'événement ;

— les 110 arguments **réduits** obtenus en prenant la forme générique pour tous les arguments à l'exception des arguments *Time, Place, Entity, Agent, Person.*

Association événement-argument

Afin d'exploiter les interdépendances entre événements, nous souhaitons modéliser le fait que la présence de certains types d'événements (*Attack*, *Die*) est informative sur la présence de certains rôles (*Target*, *Victim*). Nous ne nous restreignons pas uniquement aux liens entre un type d'événement et ses arguments spécifiques. En effet, la présence d'un événement *Attack* renforce la présence d'un argument *Victim* bien que l'événement *Attack* ne possède pas un tel argument (il contient cependant un rôle *Target*, possiblement tenu par la même mention).

Nous produisons ainsi la probabilité $p_{evt\text{-}arg}(r \mid t)$ d'observer un argument du type r sachant la présence d'un événement du type t. Nous obtenons ces probabilités au niveau de la phrase et du document à l'aide des formules suivantes :

$$p_{evt\text{-}arg\text{-}sent}(r \mid t) = \frac{num_{evt\text{-}arg\text{-}sent}(t, r)}{\sum_{r' \in R} num_{evt\text{-}arg\text{-}sent}(t, r')}$$

$$p_{\textit{evt-arg-doc}}(r \mid t) = \frac{num_{\textit{evt-arg-doc}}(t, r)}{\sum_{r' \in R} num_{\textit{evt-arg-doc}}(t, r')}$$

avec $num_{evt}(t, r)$, le nombre de fois où l'événement e et le rôle r sont observés ensemble et R, l'ensemble des rôles.

Association argument-argument

De manière similaire, si une mention d'entité joue un certain rôle (*Die-Victim*), ce fait augmente la probabilité que d'autres mentions de cette entité jouent d'autres rôles équivalents ou proches (*Attack-Target*). Pour modéliser cette dépendance, nous considérons la probabilité $p_{\textit{ent-arg}}(r2 \mid r1)$ d'observer un argument du type $r2$ sachant la présence d'un argument du type $r1$ impliquant la même entité. Nous produisons ces probabilités au niveau de la phrase et du document à l'aide des formules suivantes :

$$p_{\textit{ent-arg-sent}}(r2 \mid r1) = \frac{\sum_{e \in E} num_{\textit{ent-arg-sent}}(r1, r2, e)}{\sum_{e \in E} \sum_{r' \in R} num_{\textit{ent-arg-sent}}(r1, r', e)}$$

$$p_{\textit{ent-arg-doc}}(r2 \mid r1) = \frac{\sum_{e \in E} num_{\textit{ent-arg-doc}}(r1, r2, e)}{\sum_{e \in E} \sum_{r' \in R} num_{\textit{ent-arg-doc}}(r1, r', e)}$$

avec $num_{arg}(r1, r2, e)$, le nombre de fois où l'entité e joue à la fois les rôles $r1$ et $r2$, R, l'ensemble des rôles et E, l'ensemble des entités.

A priori locaux

Tout comme pour la prédiction des déclencheurs, nous utilisons le paramètre topPrior contrôlant le nombre de prédictions locales conservées pour les candidats (candArg). Pour chaque événement et chaque entité, nous produisons cette fois l'ensemble des cibles pour lesquelles le type de l'entité est compatible avec le rôle. Cependant, la production des a priori (candArg) est différente. Rappelons que le modèle local de Nguyen *et al.* (2016a) que nous utilisons est un modèle joint réalisant la procédure suivante :

1. attribution, pour chaque mot éligible (exclusion de certaines catégories morpho-syntaxiques), d'une probabilité associée à chaque type d'événement + classe NULLE ;
2. pour chaque mot dont la classe de probabilité maximum (i.e. le label) est un événe-

ment et non la classe NULLE :

— attribution, pour chaque entité de la phrase, d'une probabilité associée à chaque rôle + classe NULLE.

Ainsi, lorsque ce modèle local associe incorrectement la classe NULLE à un déclencheur événementiel, aucune prédiction n'est réalisée pour les arguments. De ce fait, bien qu'il soit toujours possible de prédire ces arguments, nous ne pourrons produire les observations "CandArg" correspondantes.

Avec topPrior = j, nous produisons l'ensemble des prédicats candidats et cibles selon l'heuristique décrite ci-dessous :

candidats = [], cibles = [] ;
pour *phrase* ∈ *doc* **faire**
  pour *trigger* ∈ *phrase.triggers* **faire**
    evt-type = *trigger*.evt_type ;
    pour *entité* ∈ *phrase.entités* **faire**
      ent-type = *entité*.ent_type ;
      si $pred_a(trigger, entité) \neq []$ **alors**
        candidats.ajouter("candArg(*trigger* ,*evt-type*, *entité*, 'NULLE')") ;
        pour $i \in [1..j]$ **faire**
          $arg = pred_a(trigger, entité)[i]$;
          candidat.ajouter("candArg(*trigger*, *evt-type*, *entité*, *arg*)") ;
      pour *arg* ∈ *roles*(*evt-type*) **faire**
        si *isCompatible*(*arg*, *ent-type*) **alors**
          cibles.ajouter("isArg(*trigger*, *evt-type*, *entité*, *arg*)") ;

5.6 Prédiction globale des arguments : expériences

Afin de permettre une étude juste de l'influence de notre modèle sur les performances, il est nécessaire de disposer d'un score de référence juste. L'utilisation des annotations officielles des déclencheurs implique la suppression de toutes les prédictions d'arguments associées à des faux positifs et donc nécessairement incorrectes. Cette augmentation directe de la précision rend donc la comparaison avec les performances initiales du modèle local de prédiction des arguments impossible. De plus, l'introduction de nouveaux déclencheurs (non repérés par le modèle local) nous amène, comme expliqué dans la section précédente, à produire des candidats supplémentaires pour les liens entre les entités et ces

TABLE 5.6.1 – Performances en extraction d'arguments sur la base de validation ACE 2005. Meilleure configuration pour chaque valeur du mode d'agrégation et chaque nombre d'a priori locaux

Agrégation	**P**	**R**	**F**	**std**
Modèle local	65,72	65,41	65,57	0,46
PSL	67,91	67,6	67,75	0,58
Agreg				
Réduit	67,91	67,6	67,75	0,58
Générique	67,83	67,52	67,68	0,46
Spécifique	65,85	65,55	65,7	0,45
topPrior				
1	67,91	67,6	67,75	0,58
2	67,79	67,47	67,63	0,58
3	67,75	67,44	67,59	0,36
4	67,74	67,42	67,58	0,49

nouveaux déclencheurs. Or, les prédicats impliqués dans aucune règle, comme c'est le cas pour ces cibles, se voit assigner par PSL une valeur de vérité aléatoire lors de l'inférence. De ce fait, pour ne pas surestimer les gains en rappel uniquement dus à cette instanciation aléatoire, il est nécessaire de considérer ce mécanisme pour notre modèle local. Ainsi, pour tenir compte de ces deux aspects, nous produisons un nouveau modèle local de la manière suivante : nous assignons à chaque cible la probabilité prédite par le modèle local si elle existe ou une probabilité aléatoire dans le cas contraire. Notre nouveau modèle local est ainsi obtenu en reproduisant cette expérience 100 fois et en considérant les performances moyennes.

5.6.1 Performances du modèle de base

Nous comparons en Table 5.6.1, sur le corpus de validation, le modèle local que nous venons de définir et la meilleure configuration du modèle présenté en section 5.5.2 ainsi que la meilleure configuration obtenue pour chaque valeur des paramètres d'agrégation (Agreg) et de nombre d'a priori (topPrior).

Notre modèle global obtient un gain significatif par rapport au modèle local. Ces résultats confirment notamment notre hypothèse concernant les probabilités conditionnelles des arguments spécifiques : seules les configurations utilisant l'agrégation présentent un

gain. Il n'y a toutefois pas de différence significative entre les deux modes d'intégration considérés. De plus, comme pour les expériences précédentes sur les déclencheurs, l'augmentation du nombre de candidats ne permet pas d'obtenir de meilleurs résultats.

5.6.2 Extensions du modèle de base

Afin d'améliorer les performances du modèle, nous introduisons deux nouvelles contraintes fortes visant à encoder les spécificités de la tâche.

Monoarg

Même si certains rôles peuvent apparaître plusieurs fois dans un même événement, tel que l'argument *Marry_Person* s'appliquant aux deux époux, la plupart des rôles sont uniques au sein d'un même événement. Nous modélisons ce fait à l'aide d'une contrainte forte définie ainsi :

$$\text{isArg}(w, t, +em, r) = 1\,.\,\{em : r \neq \text{'NULLE'}\}$$

Neg-Prior

Les cibles étant produites pour l'ensemble des triplets trigger-role-entité de chaque phrase, un grand nombre de cibles sont fausses : sur le jeu de validation, 70% des cibles ne sont pas réellement des arguments. Pour représenter cette information, nous introduisons la contrainte forte suivante :

$$\text{isArg}(w, t, e, \text{"NULLE"}) = 0,7 \tag{5.19}$$

La Table 5.6.2 présente les résultats de ces deux propositions et de leur combinaison sur la base de test. Notre modèle PSL initial obtient toujours des performances supérieures au modèle local, bien que le gain soit plus faible. Nous observons par ailleurs que les deux règles fournissent individuellement un gain très conséquent. Cependant, leur combinaison ne permet pas de cumuler les gains individuels.

TABLE 5.6.2 – Influence des nouvelles contraintes sur les performances du modèle PSL sur la base de test ACE 2005.

Configurations	**P**	**R**	**F**	**std**	**Gain**
Modèle local	66,37	66,07	66,22	0,49	
PSL	67,85	67,55	67,7	0,43	
+Neg-Prior	77,03	76,68	76,86	0,06	+9,16
+Monoarg	74,95	74,61	74,78	0,31	+7,08
+Monoarg +Neg-Prior	77,1	76,76	76,94	0,04	+9.24

TABLE 5.7.1 – Performances sur la base de test ACE 2005 en n'utilisant que la règle d'inférence directe. La colonne "variation" indique la différence de F-mesure par rapport au modèle utilisant toutes les règles.

Configurations	**P**	**R**	**F**	**std**	*variation*
Modèle local	66,37	66,07	66,22	0,49	-
PSL	67,64	67,33	67,48	0,57	-0,22
+NegPrior	76,99	76,38	76,68	0	-0,18
+Monoarg	74,94	74,61	74,77	0,2	-0.01

5.7 Discussion

Malgré les gains constatés, comme pour les déclencheurs, il semble qu'à l'exception de la règle d'inférence directe (eq. 5.18), les autres règles ont une influence négligeable sur les performances. La Table 5.7.1 présente les performances de notre modèle en ne conservant que les contraintes fortes et la règle d'inférence directe. Les performances obtenues dans ces conditions sont très proches de celles présentées en Table 5.6.2. La règle directe reproduisant les prédictions du modèle local, nous en concluons donc que la majorité du gain obtenu par rapport au modèle local provient des contraintes fortes. La variance très faible des modèles utilisant ces contraintes confirme également cette hypothèse.

Dans leurs travaux, Beltagy *et al.* (2014) observent que dans le cadre de la tâche de similarité textuelle, la formule de conjonction classique de PSL est trop restrictive. Liu *et al.* (2016b) font cette même observation pour l'extraction d'événements et utilisent comme Beltagy *et al.* (2014) une version modifiée de la formule de la conjonction :

$$I(l_1, l_2, \ldots, l_n) = \frac{1}{n} \sum_{i=1}^{n} I(l_i) \tag{5.20}$$

TABLE 5.7.2 – Performances sur la base de test ACE 2005 en ne considérant pas les prédictions de la classe NULLE.

Configurations	**P**	**R**	**F**	**std**
JRNN repro.	51,14	24,53	33,16	
Modèle local	28,2	73,08	40,69	0,38
PSL	28,1	75,14	40,91	0,34
+NegPrior	27,07	72,38	39,41	0,44
+Monoarg	27,1	72,45	39,45	0,49
+Monoarg + NegPrior	28,87	77,2	42,03	0,33

Contrairement à la définition originelle de la conjonction où la valeur de vérité devient nulle dès qu'un terme est nul, cette formulation permet de rendre la conjonction moins restrictive. Elle permet ainsi d'exprimer des conjonctions faisant intervenir plus de termes dont certains ont des valeurs de vérité faible. Puisque seule la règle directe semble intervenir dans l'inférence et que les autres règles comportent plus de termes, dont la probabilité conditionnelle est relativement faible, ce phénomène semble être à la source du constat que nous venons de faire sur les performances de notre modèle. Cette version alternative n'étant pas disponible dans l'implémentation officielle de PSL, nous n'avons pu y avoir recours et confirmer cette hypothèse. Dans les conditions favorables actuelles (par l'utilisation des annotations de déclencheurs), notre modèle n'est pas en mesure de tenir compte de ces dépendances. Il est donc raisonnable de supposer que l'introduction de prédictions incertaines et bruitées pour les déclencheurs ne ferait qu'amplifier ce phénomène, raison pour laquelle nous n'avons pas étudié cette modélisation.

Par ailleurs, comme nous l'avons expliqué dans la section 5.5.1, nous avons utilisé dans les précédentes expériences un programme d'évaluation considérant la classe NULLE comme une classe positive car ces conditions semblent être celles de l'article d'origine. Toutefois, il nous semble que des conditions d'évaluation plus réalistes consistent à ne calculer les performances en ne considérant que les classes correspondant à des rôles événementiels. Afin d'obtenir des résultats plus représentatifs des performances réelles du modèle, nous présentons les nouveaux résultats en Table 5.7.2, obtenus en appliquant cette méthode d'évaluation.

Dans ces conditions d'évaluation, en excluant la classe nulle, nous constatons tout

d'abord une baisse pour l'ensemble des modèles. Cette baisse se manifeste à travers la précision, du fait de notre procédure de production exhaustive des triplets {déclencheur,entité,rôle} candidats. Nous observons toutefois le maintien d'un léger gain entre le modèle local et notre modélisation PSL. Cependant, le gain observé dans l'expérience précédente par l'introduction individuelle des contraintes NegPrior et MonoArg disparaît. On observe par contre une augmentation importante des performances, due à une augmentation du rappel, lorsque les deux sont introduites ensemble.

5.8 Conclusion

Dans la continuité des contributions précédentes, nous avons proposé dans ce chapitre une modélisation visant à exploiter de manière plus globale les interactions entre les différents événements à l'échelle d'un document. Nous avons, pour ce faire, utilisé le langage d'apprentissage statistique relationnel PSL. Ce langage permet de modéliser sous forme de règles logiques un ensemble de contraintes régissant ces interdépendances. Il est alors possible de réaliser une inférence globale sur le document pour déterminer la meilleure configuration d'événements au regard de ces règles. Dans le cadre de ce chapitre nous avons réalisé deux séries d'expériences, l'une portant sur les déclencheurs événementiels et l'autre sur les arguments, toutes deux s'appuyant sur les prédictions d'un modèle neuronal local. Nous avons dans les deux cas obtenus un gain significatif par rapport aux prédictions de ce modèle local. Cependant, nous avons montré que le gain concernant les arguments n'était conséquent que dans le cadre d'évaluation supposément utilisé par (Nguyen *et al.*, 2016a) et que la pertinence de ces conditions d'évaluation était discutable. Dans les conditions d'évaluation plus rigoureuses que nous avons testées par la suite, le gain obtenu était plus faible. Par ailleurs, une étude de l'influence des différentes règles de dépendance globale a mis en évidence une incidence très faible de celles-ci sur les résultats. Par comparaison à un modèle de l'état de l'art utilisant également PSL, nous supposons que l'influence faible de ces règles est notamment due à une différence dans la sémantique de l'opérateur de conjonction empêchant l'utilisation de règles complexes faisant intervenir un nombre élevé de prédicats.

Au regard de ces différentes constatations, notre modélisation actuelle est perfectible. Néanmoins, les gains observés ainsi que ceux obtenus par le modèle de Liu *et al.* (2016b) laissent penser que la motivation première de ce chapitre est pertinente. Nous pouvons donc envisager différentes pistes d'amélioration afin d'exploiter réellement le potentiel de cette approche globale. En premier lieu, la modification de l'implémentation PSL utilisée, afin d'incorporer la redéfinition de la conjonction, permettrait de réévaluer les performances de notre modélisation actuelle. Dans le cas où cette modélisation ne serait pas satisfaisante, de nombreuses autres formulations des règles actuelles peuvent être envisagées. En particulier, une alternative à cette proposition serait d'avoir recours à un autre paradigme d'apprentissage relationnel plus mature pour implémenter notre modélisation. Nous pourrions par exemple nous inspirer des travaux de Roth et Yih (2004) formulant l'inférence jointe d'entités et de relations sous forme d'optimisation linéaire en nombres entiers (*Integer Linear Programming, ILP*).

Conclusion

Dans cette thèse, nous avons en premier lieu mis en avant les limitations des architectures neuronales classiques au regard de la portée du contexte intra-phrastique qu'elles sont en mesure d'exploiter. Nous avons alors identifié deux axes de recherche permettant de répondre à ces limitations : l'amélioration de la prise en compte du contexte intra-phrastique et l'exploitation du contexte inter-phrastique. Toutefois, la présence d'un certain nombre d'ambiguïtés au niveau intra-phrastique implique que ce contexte n'est de toute façon pas suffisant pour résoudre la tâche d'extraction d'événements. Nous avons donc choisi d'étudier la prise en compte du contexte inter-phrastique. Cette décision est par ailleurs motivée par le peu d'attention portée jusqu'à présent à ce sujet. L'exploitation de ce contexte peut être réalisée de deux manières différentes : une première approche consiste à intégrer une représentation du contexte inter-phrastique à un modèle intra-phrastique pour lui permettre de tenir compte d'informations complémentaires lors de l'analyse de la phrase courante. Cette méthode a l'avantage de pouvoir être facilement combinée aux approches neuronales actuelles. Cependant, elle fait l'hypothèse qu'en exploitant cette représentation, chaque prédiction peut être réalisée indépendamment des autres. À l'inverse, la seconde méthode de prise en compte du contexte inter-phrastique repose sur l'hypothèse qu'il existe des interdépendances fortes au niveau du document et que l'extraction correcte de l'ensemble des événements à cette échelle nécessite de résoudre ensemble les différentes ambiguïtés présentes. Cette hypothèse impose alors de réaliser une prédiction globale à l'échelle du document, ce qui n'est pas réalisable à l'aide d'architectures neuronales.

Nous avons ensuite proposé deux méthodes permettant de produire une représentation

du contexte inter-phrastique et d'intégrer cette représentation à un modèle neuronal. Enfin, nous avons également proposé deux modélisations reposant sur le langage PSL et visant à réaliser une prédiction globale des déclencheurs puis des arguments.

Résumé des travaux

Dans le chapitre 2, nous avons ré-implémenté un modèle de détection d'événements reposant sur le réseau de neurones convolutif présenté par Nguyen *et al.* (2016b). Nous avons réalisé une analyse détaillée de certains paramètres du modèle puis l'avons évalué sur plusieurs éditions de la campagne d'évaluation TAC. Nous avons également réalisé une expérience illustrant l'interdépendance des tâches d'identification et de classification des déclencheurs. Enfin, nous avons présenté une expérience comparant les performances du modèle convolutif en fonction de la taille de la fenêtre de contexte intra-phrastique autour du mot cible. Nous avons ainsi montré que les performances du modèle stagnaient à partir d'une fenêtre de 2 mots avant et après la cible alors que les modèles de l'état de l'art utilisent des fenêtres de taille 15. Nous avons déduit de cette expérience que la capacité de prise en compte des dépendances intra-phrastiques longues des modèles convolutifs était limitée, motivant ainsi notre problématique d'exploitation des dépendances inter-phrastiques.

Dans le chapitre 3, nous avons proposé une première méthode d'exploitation du contexte inter-phrastique. Cette méthode consiste à appliquer un modèle intra-phrastique au document afin de produire l'ensemble des prédictions. Ces prédictions sont ensuite agrégées par document, produisant ainsi la distribution des événements du document telle qu'estimée par le modèle local. Nous intégrons ensuite cette représentation en entrée de la couche de prédiction d'un autre modèle convolutif, lui permettant de tenir compte de cette information distante lors de la prédiction. Nous avons testé ce modèle sur l'édition 2017 de la campagne d'évaluation TAC et obtenu une augmentation de F-mesure de 0,94 point par rapport à notre modèle local, obtenant ainsi les meilleures performances pour un modèle simple. Par ailleurs, nous avons comparé cette représentation centrée sur la tâche à une représentation générique produite par doc2vec (Le et Mikolov, 2014), montrant ainsi

l'intérêt de produire une représentation propre à la tâche de détection d'événements.

Le chapitre 4 présente une nouvelle représentation de document, non seulement spécifique à la tâche, mais également à l'exemple. Nous produisons cette représentation en appliquant un modèle récurrent bidirectionnel à l'ensemble des phrases contenant des mentions d'entités en coréférence avec les entités de la phrase contenant le mot que l'on cherche à prédire. Nous entraînons ce modèle conjointement avec le modèle de détection afin de produire une représentation spécifique à la tâche. Nous avons par ailleurs reproduit le modèle de convolution de graphe proposé par Nguyen et Grishman (2018), permettant de mieux modéliser le contexte intra-phrastique. Notre réimplémentation surpasse les performances obtenues par le modèle d'origine et permet ainsi d'obtenir les meilleures performances pour un modèle simple sur les éditions 2015 et 2017 de la campagne d'évaluation TAC. Sur ces mêmes éditions, l'introduction de notre représentation permet d'obtenir respectivement des gains de 1,74 et 0,26 points en F-mesure. Nous supposons que le gain plus faible sur ce second jeu de données est dû à l'ambiguïté particulièrement forte des entités présentes dans cette collection, sur lesquelles repose notre méthode. Enfin, nous avons également comparé notre méthode à une représentation non spécifique à l'exemple (par application du modèle de contexte à l'ensemble des phrases du document) et avons ainsi montré qu'il était bénéfique de produire une représentation spécifique.

Enfin, dans le chapitre 5, nous avons considéré une autre stratégie de prise en compte du contexte inter-phrastique, reposant sur la prédiction globale des instances d'événements. Nous avons eu recours au langage de programmation statistique relationnel PSL permettant de formaliser sous forme de règles logique un ensemble de contraintes concernant les interdépendances entre événements. Ces expériences ont consisté à modifier a posteriori les prédictions du modèle local de Nguyen *et al.* (2016a) réalisant une prédiction jointe des déclencheurs et des arguments sur le jeu de données ACE 2005. Nous avons dans un premier temps considéré la prédiction globale des déclencheurs et pu obtenir un gain de 0,92 point en test. Toutefois, l'influence des différentes règles d'interaction globale s'est révélée marginale. Dans un second temps, nous avons étudié la prédiction des arguments en considérant les déclencheurs comme acquis. Nous avons également obtenu

un gain, cette fois relativement faible, et de nouveau constaté le peu d'influence des différentes règles. Nous avons supposé que ce phénomène était dû à une limitation du pouvoir expressif des règles dans le langage PSL.

Perspectives

Nous commençons ici par résumer les perspectives spécifiques aux deux familles d'approches globales considérées. Concernant l'exploitation du contexte par extraction et intégration d'une représentation vectorielle, l'approche présentée dans le chapitre 4 nous semble la plus pertinente, puisqu'elle est à la fois spécifique à la tâche et à l'exemple et que la représentation est entraînée en même temps que le modèle de classification. Afin d'améliorer ses performances, fortement dépendantes de la détection des entités et de leurs coréférences, l'utilisation d'outils de désambiguïsation d'entités (*Entity Linking*) nous semble la piste la plus prometteuse. Par ailleurs, l'utilisation d'un mécanisme d'attention pour filtrer les informations du contexte distant pourrait s'avérer utile.

Concernant les approches de prédiction globale, nos expériences avec PSL n'ont pas permis de tenir compte des dépendances inter-phrastiques à travers l'optimisation globale des contraintes. Ce paradigme nous semble cependant pertinent au vu des bons résultats obtenus par ces approches d'optimisation de contraintes globales, par exemple pour la désambiguïsation collective des entités d'un texte ou en résumé automatique. Il serait donc intéressant de considérer d'autres méthodes de programmation statistique relationnelle ainsi que différentes formalisations de contraintes inter-phrastiques.

Enfin, ces deux approches ne considère pas ensemble et au même niveau de finesse les contextes intra-phrastique et inter-phrastique. Dans le cas des approches par représentation, seule une version condensée du contexte inter-phrastique est considérée lors de la prédiction, alors que le contexte intra-phrastique est exploité avec plus de finesse. Dans le cas de l'approche par prédiction globale, c'est le contexte intra-phrastique, d'abord exploité finement qui est ensuite abstrait et condensé sous la forme d'a priori locaux pour être ensuite exploité au niveau inter-phrastique. Une perspective évidente consisterait donc à réaliser un modèle permettant d'exploiter finement et conjointement ces deux contextes.

Cependant, si cette perspective est aisément identifiée, une modélisation techniquement viable ne nous semble pas envisageable à l'heure actuelle.

De manière plus générale, la notion de contexte pourrait être enrichie selon deux axes différents. D'une part, nous nous sommes limité, jusqu'à présent à l'analyse syntaxique et sémantique des documents, notamment à travers l'exploitation des liens de coréférence. L'exploitation de la structure discursive du document, telle qu'envisagée dans le cadre de la RST (*Rhetorical Structure Theory*, (Mann et Thompson, 1988)) par exemple, offre la potentialité de mieux modéliser les interdépendances à l'échelle d'un document. Elle pourrait ainsi permettre d'identifier, à travers la structure discursive du document, les phrases les plus pertinentes au sein du contexte inter-phrastique. Par ailleurs, la notion de contexte, jusqu'ici restreinte au niveau d'un document pourrait être également étendue au-delà du document. Cette idée a déjà été explorée dans Ji et Grishman (2008) où les auteurs utilisent les prédictions faites sur un document de test pour produire une requête permettant d'identifier des documents similaires afin de corriger les prédictions initiales, obtenant ainsi un gain très important. Ces corrections s'appuient sur un ensemble de règles visant à maximiser la cohérence globale des événements à l'échelle du document et du corpus. Toutefois, cette méthode repose sur un modèle local dont les performances sont relativement faibles comparées aux architectures récentes. L'application de cette méthode aux prédictions d'un modèle neuronal récent serait intéressante du point de vue théorique car l'étude de l'influence de ces différentes règles permettrait de mieux identifier les proportions de cas où la phrase d'une part et le document d'autre part ne constituent pas des contextes suffisants pour résoudre la tâche. Par ailleurs, nous pourrions également utiliser la désambiguïsation d'entités, évoquée comme perspective dans le cas de notre représentation dynamique du contexte, pour identifier des documents externes contenant les mêmes entités afin d'étendre le contexte considéré.

Sur le plan méthodologique, la plupart des travaux publiés sur la tâche d'extraction d'événements mettent l'accent sur la proposition de nouvelles méthodes obtenant des gains plus ou moins importants. Ces résultats sont ainsi peu informatifs sur l'apport spécifique de la contribution proposée. Cette incertitude est renforcée par l'absence systématique

des performances moyennes et des variances, ces dernières pouvant être plus importantes que les gains mis en avant par ces articles, ce qui a été mis en évidence par les travaux de Orr *et al.* (2018). De ce fait, la perspective nous semblant la plus pertinente serait le développement d'études analytiques visant à mieux appréhender la contribution spécifique des différents choix de modélisation sur les performances de détection d'événements. Une première déclinaison de cette perspective pourrait concerner l'étude systématique de la taille des dépendances intra-phrastiques exploitables par les modèles convolutifs, récurrents et de graphe, dans la prolongation de celle présentée dans le chapitre 2. De telles études permettraient ainsi de mieux comprendre les limites qualitatives de ces différentes approches et donc d'identifier des axes de recherche prometteurs. Ce type d'étude est cependant difficile à l'heure actuelle, du fait du nombre très restreint de modèles de l'état de l'art dont le code est disponible.

Par ailleurs, l'introduction de l'architecture *Transformer* (Vaswani *et al.*, 2017) dans le cadre de la traduction automatique a permis d'améliorer les performances de l'état de l'art tout en réduisant fortement le coût d'apprentissage nécessaire, comparé aux modèles antérieurs. Cette architecture, reposant sur la prédiction de séquence, pourrait ainsi être appliquée pour une prédiction globale des événements de la phrase. Cependant, les données disponibles en extraction d'événements étant limitées, il n'est pas certain que cette architecture complexe soit directement entraînable sur cette tâche. Cependant, les développements récents de modèles de langue neuronaux profonds tels que ULMFiT (Howard et Ruder, 2018), ELMO (Peters *et al.*, 2018) ou BERT (Devlin *et al.*, 2018) (utilisant une architecture *Transformer*), dont des versions pré-entraînées sont disponibles, permet d'envisager des approches par adaptation de modèles (*Transfer Learning*). Ceci permettrait de tirer parti de ces architectures complexes tout en atténuant le besoin de données d'entraînement spécifiques.

Publications

Plusieurs des contributions présentées dans ce manuscrit ont été publiées dans des journaux ou des conférences à comité de lecture, nous les présentons ici.

Une première version courte de l'état de l'art présenté dans le chapitre 1 a été publiée dans l'édition 2017 des Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA) (Kodelja *et al.*, 2017a). Une version étendue de cet état de l'art a été acceptée dans la revue TAL (Kodelja *et al.*, 2019b).

Les expériences du chapitre 2 ont été réalisées dans le cadre de la campagne d'évaluation TAC 2017 pour laquelle nous avons produit un rapport technique (Kodelja *et al.*, 2017b).

Les travaux présentés dans le chapitre 3 ont été publiés dans un article long présenté lors de l'édition 2018 de la conférence sur le Traitement Automatique des Langues Naturelles (TALN) (Kodelja *et al.*, 2018) ainsi que dans un article court de l'édition 2019 de la *European Conference on Information Retrieval* (ECIR) (Kodelja *et al.*, 2019a).

Bibliographie

Omri Abend et Ari Rappoport : The State of the Art in Semantic Representation. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 77–89, Vancouver, Canada, 2017. Association for Computational Linguistics.

David Ahn : The Stages of Event Extraction. *In Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia, 2006. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho et Yoshua Bengio : Neural machine translation by jointly learning to align and translate. *In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, CA, USA, mai 2015.

Collin F. Baker, Charles J. Fillmore et John B. Lowe : The Berkeley FrameNet Project. *In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1,*, pages 86–90, Montréal, Quebec, Canada, 1998.

Marco Baroni, Georgiana Dinu et Germán Kruszewski : Don't count, predict ! A systematic comparison of context-counting vs. context-predicting semantic vectors. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 238–247, Baltimore, Maryland, juin 2014. Association for Computational Linguistics.

Islam Beltagy, Katrin Erk et Raymond Mooney : Probabilistic Soft Logic for Semantic Textual Similarity. *In Proceedings of the 52nd Annual Meeting of the Association*

for Computational Linguistics (Volume 1 : Long Papers), pages 1210–1219, Baltimore, Maryland, juin 2014. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent et Christian Jauvin : A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003.

David M Blei, Andrew Y Ng et Michael I Jordan : Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, 2003.

Piotr Bojanowski, Edouard Grave, Armand Joulin et Tomas Mikolov : Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge et Jamie Taylor : Freebase : a collaboratively created graph database for structuring human knowledge. *In Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, New York, NY, USA, 2008. ACM.

Emanuela Boroş : *Neural Methods for Event Extraction.* Thèse de doctorat, Université Paris-Saclay, 2018.

Ofer Bronstein, Ido Dagan, Qi Li, Heng Ji et Anette Frank : Seed-Based Event Trigger Labeling : How far can event descriptions get us ? *In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers)*, volume 2, pages 372–376, 2015.

Chen Chen et Vincent Ng : Joint Modeling for Chinese Event Extraction with Rich Linguistic Features. *In Proceedings of the 24th International Conference on Computational Linguistics*, pages 529–544, Mumbai, India, 2012. ACL.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu et Jun Zhao : Automatically Labeled Data Generation for Large Scale Event Extraction. *In Proceedings of the 55th Annual*

Meeting of the Association for Computational Linguistics, pages 409–419, Vancouver, Canada, 2017.

Yubo CHEN, Liheng XU, Kang LIU, Daojian ZENG et Jun ZHAO : Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. *In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 167–176, Beijing, China, 2015. ACL.

Zheng CHEN et Heng JI : Language Specific Issue and Feature Exploration in Chinese Event Extraction. *In Proceedings of Human Language Technologies : The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-Short '09, pages 209–212, Stroudsburg, PA, USA, 2009. ACL.

Kyunghyun CHO, Bart van MERRIENBOER, Caglar GULCEHRE, Dzmitry BAHDANAU, Fethi BOUGARES, Holger SCHWENK et Yoshua BENGIO : Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation . *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1724–1734, Doha, Qatar, 2014. ACL.

Ronan COLLOBERT et Jason WESTON : A Unified Architecture for Natural Language Processing : Deep Neural Networks with Multitask Learning. *In Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM.

Ronan COLLOBERT, Jason WESTON, Léon BOTTOU, Michael KARLEN, Koray KAVUKCUOGLU et Pavel KUKSA : Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

Jacob DEVLIN, Ming-Wei CHANG, Kenton LEE et Kristina TOUTANOVA : Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv :1810.04805*, 2018.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel et Ralph Weischedel : The Automatic Content Extraction (ACE) Program – Tasks, Data, and Evaluation. *In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, mai 2004. European Language Resources Association (ELRA).

Shaoyang Duan, Ruifang He et Wenli Zhao : 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, pages 352–361, 2017.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin et Ting Liu : A Language-Independent Neural Network for Event Detection. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, volume 2, pages 66–71, 2016.

John R Firth : A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, 1957.

Michael R Genesereth et Nils J Nilsson : Logical foundations of Artificial Intelligence. *Morgan Kaufmann*, 2, 1987.

Jeremy Getman, Joe Ellis, Stephanie Strassel, Zhiyi Song et Jennifer Tracey : 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, mai 2018. European Languages Resources Association (ELRA).

Kevin Gimpel et Noah Smith : Softmax-Margin CRFs : Training Log-Linear Models with Cost Functions. *In Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 733–736, 2010.

Christoph GOLLER et Andreas KÜCHLER : Learning Task-Dependent Distributed Representations by Backpropagation Through Structure. *In Proceedings of International Conference on Neural Networks (ICNN'96)*, pages 347–352. IEEE, 1996.

David GRAFF et Christopher CIERI : English gigaword, ldc catalog no. *LDC2003T05. Linguistic Data Consortium, University of Pennsylvania*, 2003.

Ralph GRISHMAN : Twenty-five years of information extraction. *Natural Language Engineering*, pages 1–16, 2019.

Ralph GRISHMAN et Beth SUNDHEIM : Message Understanding Conference-6 : A Brief History. *In Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

Ralph GRISHMAN, David WESTBROOK et Adam MEYERS : NYU's English ACE 2005 System Description. *In Proceedings of ACE 2005 Evaluation Workshop*, volume 5, 2005.

Zellig S HARRIS : Distributional Structure. *Word*, 10(2-3):146–162, 1954.

Jerry R. HOBBS : Overview of the TACITUS Project. *the finite string newsletter*, 12 (3):220–222, 1986.

Jerry R. HOBBS, Douglas E. APPELT, John BEAR, David ISRAEL, Megumi KAMEYAMA, Marc STICKEL et Mabry TYSON : FASTUS : A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. *Finite-State Language Processing*, pages 383–406, 1997.

Sepp HOCHREITER et Jürgen SCHMIDHUBER : Long Short-Term Memory. *Neural Computation*, 9(9):1735–1780, 1997.

Yu HONG, Di LU, Dian YU, Xiaoman PAN, Xiaobin WANG, Yadong CHEN, Lifu HUANG et Heng JI : RPI_BLENDER TAC-KBP2015 System Description. *In Proceedings of the 2015 Text Analysis Conference*, 2015.

Yu HONG, Jianfeng ZHANG, Bin MA, Jianmin YAO, Guodong ZHOU et Qiaoming ZHU : 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, pages 1127–1136. ACL, 2011.

Yu HONG, Wenxuan ZHOU, Jingli ZHANG, Qiaoming ZHU et GuoDong ZHOU : Self-Regulation : Employing a Generative Adversarial Network to Improve Event Detection. *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, 2018. ACL.

Jeremy HOWARD et Sebastian RUDER : Universal Language Model Fine-tuning for Text Classification. *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 328–339, Melbourne, Australia, juillet 2018.

Ludovic JEAN-LOUIS, Romaric BESANÇON et Olivier FERRET : Text Segmentation and Graph-based Method for Template Filling in Information Extraction. *In Proceedings of the 5^{th} International Joint Conference on Natural Language Processing (IJCNLP 2011)*, pages 723–731, Chiang Mai, Thailand, 2011.

Heng JI et Ralph GRISHMAN : Refining Event Extraction through Cross-Document Inference. *Association for Computational Linguistics*, pages 254–262, 2008.

Shanshan JIANG, Yihan LI, Tianyi QIN, Qian MENG et Bin DONG : SRCB Entity Discovery and Linking (EDL) and Event Nugget Systems for TAC 2017. *In Proceedings of the 2017 Text Analysis Conference*, 2017.

Armand JOULIN, Edouard GRAVE, Piotr BOJANOWSKI et Tomas MIKOLOV : Bag of Tricks for Efficient Text Classification. *In Proceedings of the 15th Conference of the European Chapter of the Association for Computational (Short Papers)*, pages 427–431, Valencia, Spain, 2017. ACL.

Rafal JOZEFOWICZ, Wojciech ZAREMBA et Ilya SUTSKEVER : An Empirical Exploration of Recurrent Network Architectures. *In Proceedings of the 32nd International Confe-*

rence on International Conference on Machine Learning, volume 37 de *ICML'15*, pages 2342–2350, Lille, France, 2015. JMLR.

Thomas N. Kipf et Max Welling : 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*, 2017.

Dorian Kodelja, Romaric Besançon et Olivier Ferret : Représentations et modèles en extraction d'événements supervisée. *In Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA 2017)*, Caen, France, 2017a.

Dorian Kodelja, Romaric Besançon et Olivier Ferret : Intégration de Contexte Global Par Amorçage pour la Détection d'événements. *In 25ème Conférence Sur Le Traitement Automatique Des Langues Naturelles*, Rennes, France, 2018. ATALA.

Dorian Kodelja, Romaric Besançon et Olivier Ferret : Exploiting a More Global Context for Event Detection Through Bootstrapping. *In Proceedings of the 41st European Conference on Information Retrieval* , pages 763–770, 2019a.

Dorian Kodelja, Romaric Besançon, Olivier Ferret, Hervé Le Borgne et E. Boroş : CEA LIST Participation to the TAC 2017 Event Nugget Track. *In Proceedings of the 10th Text Analysis Conference (TAC 2017)*, Gaithersburg, United States, 2017b.

Dorian Kodelja, Romaric Besançon et Olivier Ferret : Modèles neuronaux pour l'extraction supervisée d'événements : état de l'art. *TAL*, 60(1):13–37, 2019b.

John D. Lafferty, Andrew McCallum et Fernando C. N. Pereira : Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data. *In Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

Quoc Le et Tomas Mikolov : 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*, pages 1188–1196, Beijing, China, 2014. JMLR.org.

Omer Levy, Yoav Goldberg et Ido Dagan : Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

Qi Li, Heng Ji, Yu Hong et Sujian Li : Constructing Information Networks Using One Single Model. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1846–1851, Doha, Qatar, 2014. ACL.

Qi Li, Heng Ji et Liang Huang : Joint Event Extraction via Structured Prediction with Global Features. *In Proceeding of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 73–82, Sofia, Bulgaria, 2013. ACL.

Shasha Liao et Ralph Grishman : Using Document Level Cross-Event Inference to Improve Event Extraction. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden, 2010. ACL.

Wang Ling, Chris Dyer, Alan W Black et Isabel Trancoso : Two/Too Simple Adaptations of Word2Vec for Syntax Problems. *In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 1299–1304, Denver, Colorado, mai 2015. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu et Jun Zhao : Event Detection via Gated Multilingual Attention Mechanism. *In Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, 2018. AAAI Press.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu et Jun Zhao : Leveraging FrameNet to Improve Automatic Event Detection. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 2134–2143, Berlin, Germany, août 2016a. Association for Computational Linguistics.

Shulin Liu, Yubo Chen, Kang Liu et Jun Zhao : Exploiting Argument Information to Improve Event Detection via Supervised Attention Mechanisms. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1789–1798, Vancouver, Canada, 2017. ACL.

Shulin Liu, Kang Liu, Shizhu He et Jun Zhao : 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, 2016b. AAAI Press.

Zhengzhong Liu, Jun Araki, Teruko Mitamura et Eduard H Hovy : CMU-LTI at KBP 2016 Event Nugget Track. *In Proceedings of the 2016 Text Analysis Conference*, 2016c.

Jing Lu et Vincent Ng : UTD's Event Nugget Detection and Coreference System at KBP 2016. *In Proceedings of the 2016 Text Analysis Conference*, 2016.

Steven L. Lytinen et Anatole Gershman : ATRANS Automatic Processing of Money Transfer Messages. *In Proceedings of the 5th National Conference of the American Association for Artificial Intelligence*, volume 86, pages 1089–1093, Philadelphia, PA, USA, 1986. AAAI Press.

Peter Makarov et Simon Clematide : UZH at TAC KBP 2017 : Event Nugget Detection via Joint Learning with Softmax-Margin Objective. *In Proceedings of the 2017 Text Analysis Conference*, 2017.

William C Mann et Sandra A Thompson : Rhetorical structure theory : Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard et David McClosky : The Stanford CoreNLP Natural Language Processing Toolkit. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), System Demonstrations*, pages 55–60, 2014.

Joseph MARINO : Taxonomic Curriculum Learning for Fine-Grained Object Recognition. *Semantic Scholar*, 2016.

Tomas MIKOLOV, Ilya SUTSKEVER, Kai CHEN, Greg S. CORRADO et Jeffrey DEAN : Distributed Representations of Words and Phrases and Their Compositionality. *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

George A. MILLER : WordNet : A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

Teruko MITAMURA, Zhengzhong LIU et Eduard HOVY : Overview of TAC KBP 2016 Event Nugget Track. *In Proceedings of the 2016 Text Analysis Conference*, 2016.

Teruko MITAMURA, Zhengzhong LIU et Eduard HOVY : Events Detection, Coreference and Sequencing : What's next ? Overview of the TAC KBP 2017 Event Track. *In Proceedings of the 2017 Text Analysis Conference*, 2017.

Teruko MITAMURA, Yukari YAMAKAWA, Susan HOLM, Zhiyi SONG, Ann BIES, Seth KULICK et Stephanie STRASSEL : Event Nugget Annotation : Processes and Issues. *In Proceedings of the The 3rd Workshop on EVENTS : Definition, Detection, Coreference, and Representation*, pages 66–76, Denver, Colorado, 2015.

Sean MONAHAN, Michael MOHLER, Marc TOMLINSON, Amy BOOK, Maxim GORELKIN, Kevin CROSBY et Mary BRUNSON : Populating a Knowledge Base with Information about Events. *In Proceedings of the 2015 Text Analysis Conference*, 2015.

Raymond J. MOONEY et Mary E. CALIFF : Relational Learning of Pattern-Match Rules for Information Extraction. *In Proceedings of the Sixteenth National Conference on Artificial Intelligence*, volume 328, page 334, Orlando, FL, USA, 1999. AAAI Press.

Thien H. NGUYEN, Kyunghyun CHO et Ralph GRISHMAN : Joint Event Extraction via Recurrent Neural Networks. *In Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 300–309, San Diego, CA, USA, 2016a. ACL.

Thien H. Nguyen et Ralph Grishman : Employing Word Representations and Regularization for Domain Adaptation of Relation Extraction. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 68–74, Baltimore, MD, USA, 2014. ACL.

Thien H. Nguyen et Ralph Grishman : Event Detection and Domain Adaptation with Convolutional Neural Networks. *In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 2, pages 365–371, Beijing, China, 2015a. ACL.

Thien H. Nguyen et Ralph Grishman : Relation Extraction : Perspective from Convolutional Neural Networks. *In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 39–48, Denver, CO, USA, 2015b. ACL.

Thien Huu Nguyen et Ralph Grishman : Modeling Skip-Grams for Event Detection with Convolutional Neural Networks. *In Proceedings of the 2016 Conference on Empirical Methods on Natural Language Processing*, Austin, TX, USA, 2016. ACL.

Thien Huu Nguyen et Ralph Grishman : 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, 2018. AAAI Press.

Thien Huu Nguyen, Ralph Grishman et Adam Meyers : New York University 2016 System for KBP Event Nugget : A Deep Learning Approach. *In Proceedings of the 2016 Text Analysis Conference*, Gaithersburg, MD, USA, 2016b. NIST.

J. Walker Orr, Prasad Tadepalli et Xiaoli Fern : Event Detection with Neural Networks : A Rigorous Empirical Evaluation. *In Proceedings of the 2018 Conference on Empirical Methods on Natural Language Processing*, Brussels, Belgium, 2018. ACL.

Sachin Pawar, Pushpak Bhattacharya et Girish K. Palshikar : End-to-end relation extraction using markov logic networks. *In Computational Linguistics and Intelligent Text Processing*, pages 535–551, 2018.

Jeffrey Pennington, Richard Socher et Christopher Manning : Glove : Global Vectors for Word Representation. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar, 2014. ACL.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee et Luke Zettlemoyer : Deep Contextualized Word Representations. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 2227–2237, New Orleans, Louisiana, 2018. ACL.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei et Ilya Sutskever : Language Models are Unsupervised Multitask Learners. 2019.

Lisa F. Rau : Conceptual Information Extraction from Financial News. *In Proceedings of the 21st Annual Hawaii International Conference on System Sciences*, volume 3, pages 501–509, Kailua-Kona, HI, USA, 1988. IEEE.

Nils Reimers et Iryna Gurevych : Reporting Score Distributions Makes a Difference : Performance Study of LSTM-networks for Sequence Tagging. *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 338–348, Copenhagen, Denmark, 2017. ACL.

Matthew Richardson et Pedro Domingos : Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

Ellen Riloff : Automatically Constructing a Dictionary for Information Extraction Tasks. *In Proceedings of the 11th National Conference of the American Association for Artificial Intelligence*, pages 811–816, Washington, D.C., USA, 1993. AAAI Press.

Marco Rospocher : An ontology-driven probabilistic soft logic approach to improve nlp entity annotations. *In The Semantic Web – ISWC 2018*, pages 144–161, 2018.

Dan Roth et Wen-tau Yih : A linear programming formulation for global inference in natural language tasks. *In Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, 2004.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov et Max Welling : Modeling relational data with graph convolutional networks. *In Proceedings of the 15th European Semantic Web Conference*, pages 593–607. Springer, 2018.

Lei Sha, Feng Qian, Baobao Chang et Zhifang Sui : Jointly Extracting Event Triggers and Arguments by Dependency-Bridge RNN and Tensor-Based Argument Interaction. *In Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, 2018. AAAI Press.

Richard Socher : *Recursive deep learning for natural language processing and computer vision.* Thèse de doctorat, Citeseer, 2014.

Mark Stevenson : Fact Distribution in Information Extraction. *Language Resources and Evaluation*, 40(2):183–201, 2006.

Ang Sun, Ralph Grishman et Satoshi Sekine : Semi-Supervised Relation Extraction with Large-Scale Word Clustering. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, volume 1, pages 521–529, Portland, OR, USA, 2011. ACL.

Youssef Tamaazousti, Hervé Le Borgne, Céline Hudelot, Mohamed El Amine Seddik et Mohamed Tamaazousti : Learning More Universal Representations for Transfer-Learning. *arXiv :1712.09708 [cs]*, 2017.

Tatiana Tommasi, Novi Patricia, Barbara Caputo et Tinne Tuytelaars : A deeper look at dataset bias. *In Domain Adaptation in Computer Vision Applications*, pages 37–55. Springer, 2017.

Joseph Turian, Lev Ratinov et Yoshua Bengio : Word Representations : A Simple and General Method for Semi-Supervised Learning. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA, 2010. ACL.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser et Illia Polosukhin : Attention is all you need. *In Advances in Neural Information Processing Systems 30*, pages 5998–6008. 2017.

Bishan Yang et Tom M. Mitchell : 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*, pages 289–299, San Diego, California, 2016. ACL.

Roman Yangarber et Lauri Jokipii : Redundancy-Based Correction of Automatically Extracted Facts. *In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 57–64, Stroudsburg, PA, USA, 2005. ACL.

Daojian Zeng, Kang Liu, Yubo Chen et Jun Zhao : Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal, 2015. ACL.

Jingli Zhang, Wenxuan Zhou, Yu Hong, Jianmin Yao et Min Zhang : Using Entity Relation to Improve Event Detection via Attention Mechanism. *In Proceedings of the Seventh CCF International Conference on Natural Language Processing and Chinese Computing*, Lecture Notes in Computer Science, pages 171–183, Hohhot, China, 2018. Springer.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang et Xueqi Cheng : 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)*, pages 414–419, Melbourne, Australia, 2018. ACL.

GuoDong Zhou, Jian Su, Jie Zhang et Min Zhang : Exploring Various Knowledge in Relation Extraction. *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 427–434, Ann Arbor, MI, USA, 2005. ACL.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Prise en compte du contexte inter-phrastique pour l'extraction d'événements supervisée

Mots clés : Traitement automatique des langues, extraction d'information, apprentissage automatique

Résumé : Un des principaux pans du traitement automatique des langues (TAL) est l'extraction sous forme structurée des informations contenues dans un document. Cette extraction est généralement constituée de trois étapes : l'extraction d'entités nommées, des relations les liant et enfin celle des événements. Cette étape est communément considérée comme la plus difficile des trois. La notion d'événement recouvre différents phénomènes caractérisés par un nombre variable d'actants. L'extraction d'événements consiste alors à identifier la présence d'un événement puis à en déterminer les arguments, c'est-à-dire les différentes entités y remplissant des rôles spécifiques. Ces deux étapes sont généralement traitées successivement et la première étape repose alors sur la détection d'un déclencheur indiquant la présence d'un événement. Les meilleures approches actuelles, reposant sur des modèles neuronaux, se focalisent sur le voisinage direct du mot dans la phrase. Les informations présentes dans le reste du document sont alors ignorées. Cette thèse présente différentes approches visant à exploiter ce contexte distant . Nous reproduisons en premier lieu un modèle convolutif obtenant des performances à l'état de l'art et présentons le fait que ce modèle, malgré ses bonnes performances, n'exploite effectivement qu'un contexte très restreint au niveau phrastique. Dans un deuxième temps, nous présentons deux méthodes de production et d'intégration d'une représentation du contexte distant à un modèle neuronal opérant au niveau intra-phrastique. La première contribution se fonde sur un mécanisme d'amorçage en produisant une représentation du document spécifique à la tâche par agrégation des prédictions d'un premier modèle intra-phrastique puis en l'intégrant à un nouveau modèle. Une seconde contribution, répondant aux limitations de la première méthode, permet d'exploiter dynamiquement, pour chaque cible de prédiction, une représentation des phrases les plus pertinentes au sein du contexte grâce à un modèle de convolution de graphe. Enfin, dans un troisième temps, nous considérons une autre approche de la prise en compte du contexte inter-phrastique. Nous cherchons à modéliser plus directement les interdépendances entre les différentes instances d'événements au sein d'un document afin de réaliser une prédiction jointe. Nous utilisons pour cela le cadre d'apprentissage PSL (*Probabilistic Soft Logic*) qui permet de modéliser de telles interdépendances sous forme de règles logiques.

Title : Leveraging cross-sentential context for supervised event extraction

Keywords : Natural Language Processing, information extraction, machine learning

Abstract : The extraction of structured information from a document is one of the main parts of natural language processing (NLP). This extraction usually consists in three steps: named entities recognition relation extraction and event extraction. This last step is considered to be the most challenging. The notion of event covers a broad list of different phenomena which are characterized through a varying number of roles. Thereupon, Event extraction consists in detecting the occurrence of an event then determining its argument, that is, the different entities filling specific roles. These two steps are usually done one after the other. In this case, the first step revolves around detecting triggers indicating the occurrence of events. The current best approaches, based on neural networks, focus on the direct neighborhood of the target word in the sentence. Information in the rest of the document is then usually ignored. This thesis presents different approaches aiming at exploiting this document-level context. We begin by reproducing a state of the art convolutional neural network and showing that, despite its good performances, it only exploit a narrow context at the intra-sentential level. Subsequently, we present two methods to generate and integrate a representation of the inter-sentential context in a neural network operating on an intra-sentential context. The first contribution consists in producing a task-specific representation of the inter-sentential context through the aggregation of the predictions of a first intra-sentential model. Our second contribution, in response to the limitations of the first one, allows for the dynamic generation of a specific context for each target word. Finally, we take a different tack on the exploitation of the inter-sentential context. We try a more direct modelisation of the dependencies between multiple event instances inside a document in order to produce a joint prediction. To do so, we use the PSL (*Probabilistic Soft Logic*) framework which allows to model such dependencies through logic formula.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France