Université de Montréal

Utilisation de représentations de mots pour l'étiquetage de rôles sémantiques suivant FrameNet

par
William Léchelle

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Janvier, 2014

RÉSUMÉ

Dans la sémantique des cadres de Fillmore, les mots prennent leur sens par rapport au contexte événementiel ou situationnel dans lequel ils s'inscrivent. FrameNet, une ressource lexicale pour l'anglais, définit environ 1000 cadres conceptuels, couvrant l'essentiel des contextes possibles.

Dans un cadre conceptuel, un prédicat appelle des arguments pour remplir les différents rôles sémantiques associés au cadre (par exemple : `Victime`, `Manière`, `Receveur`, `Locuteur`). Nous cherchons à annoter automatiquement ces rôles sémantiques, étant donné le cadre sémantique et le prédicat.

Pour cela, nous entrainons un algorithme d'apprentissage machine sur des arguments dont le rôle est connu, pour généraliser aux arguments dont le rôle est inconnu. On utilisera notamment des propriétés lexicales de proximité sémantique des mots les plus représentatifs des arguments, en particulier en utilisant des représentations vectorielles des mots du lexique.

Mots-clefs : traitement des langues ; apprentissage supervisé ; rôles sémantiques ; FrameNet ; représentations distribuées

ABSTRACT

According to Frame Semantics (Fillmore 1976), word meanings are best understood considering the semantic frame they play a role in, for the frame is what gives them context. FrameNet is a lexical database that defines about 1000 semantic frames, along with the roles to be filled by arguments to the predicate calling the frame in a sentence. Our task is to automatically label argument roles, given their position, the frame, and the predicate (sometimes refered to as semantic role labelling).

For this task, I make use of distributed word representations, in order to improve generalisation over the few training exemples available for each frame. A maximum entropy classifier using common features of the arguments is used as a strong baseline to be improved upon.

Keywords : natural language processing ; supervised learning ; semantic role labelling ; FrameNet ; word representations

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

CHAPITRE 1

INTRODUCTION

Le traitement automatique des langues vise à construire une représentation formelle du langage humain qui soit compréhensible et donc manipulable par une machine. Cela permet notamment de résumer, traduire, ou extraire de l'information automatiquement à partir de données textuelles.

Ce mémoire se penche sur une représentation de l'information sémantique contenue dans le texte : on fait correspondre des phrases à des évènements ou à des situations mettant en rapport des agents et des facteurs, chacun de ces éléments devant correspondre à une partie du texte clairement identifiée.

Par exemple, dans la phrase *Je me promène dans le bois*, il s'agit d'identifier le mouvement (*promène*) d'un agent (*Je*), précisé de la zone dans laquelle a lieu ce mouvement (*dans le bois*). On peut identifier plus précisément que c'est un mouvement intentionnel de l'agent lui-même, par opposition à *Je tombe dans le puits*, par exemple.

La recherche de cadres typiques descriptifs de l'information sémantique est motivée par la sémantique des cadres, introduite en section 1.1. L'ensemble des évènements ou situations qu'on peut chercher à identifier est défini et décrit par FrameNet, présenté section 1.2. Je présente ensuite la tâche d'analyse automatique, et les ressources complémentaires que j'ai employées (des représentations distribuées de mots) pour améliorer la réalisation de cette tâche, en généralisant les données d'apprentissage par rapport au lexique.

La section 2 présente l'état de l'art en étiquetage sémantique. Je décris les expériences que j'ai menées, et présente et détaille leurs résultats dans la section 3, avant de conclure.

1.1 Sémantique des cadres

La sémantique des cadres est une théorie développée par Fillmore dans les années 70 (voir par exemple Fillmore 1976), qui stipule que la compréhension du sens d'un mot passe avant tout par la connaissance du cadre dans lequel ce mot est utilisé.

Par exemple, le mot *acheter* tient son sens du contexte de l'échange commercial entre de la monnaie et un bien, impliquant un acheteur et un vendeur, une méthode de transaction, etc. De même, *vente* ou *cher* renvoient au même cadre général, mais mettent l'accent sur un contexte plus restreint du scénario commercial.

1.2 FrameNet

Développé depuis 1997 à l'université Berkeley, le projet FrameNet [1] vise à construire une base de données lexicale pour l'anglais, lisible par la machine, pour décrire la façon dont les mots peuvent se combiner, en fonction de leur cadre sémantique.

La ressource peut se décomposer en 3 parties que je détaillerai tour à tour :

- la description des cadres sémantiques ;
- le dictionnaire d'unités lexicales ;
- le corpus annoté.

Dans la suite du texte, je traduis l'essentiel des exemples en français, mais FrameNet étant bien une ressource construite pour l'anglais, j'ai dû garder un minimum d'éléments dans la langue originale.

J'ai utilisé la version 1.5 de FrameNet [2] pour mener mes expériences.

1. `https://framenet.icsi.berkeley.edu/fndrupal/`
2. publiée en septembre 2010, j'ai utilisé une version téléchargée en février 2013

1.2.1 Cadres sémantiques

FrameNet s'appuie sur la sémantique des cadres de Fillmore (voir section 1.1), et définit ainsi un peu plus de 1000 cadres, visant à couvrir tous les évènements ou tous les contextes possibles, au niveau général. Un cadre sémantique consiste en une courte description (en anglais) du contexte situationnel ou de l'évènement considéré, d'une liste de rôles sémantiques pouvant être remplis dans ce cadre, et d'une liste d'unités lexicales qui appellent ce cadre dans le texte. Certains rôles plus fondamentaux à la situation sont dits appartenir au noyau du cadre (*Core Frame Elements*).

Par exemple, le cadre `Communication`[3] est décrit comme suit (où les rôles sont mis en évidence) :

> Un `Communicateur` transmet un `Message` à un `Destinataire` ; le `Sujet` et `Medium` de la communication pouvant aussi être exprimés. Ce cadre ne spécifie pas la méthode de communication (oral, écrit, geste, etc.). Les cadres qui héritent de ce cadre général de `Communication` peuvent ajouter des détails au `Medium` de différentes façons (en français, à la radio, dans une lettre), ou à la `Façon` de communiquer (bavardage, diatribe, cri, murmure).

Les rôles du noyau de ce cadre sont `Communicateur`, `Medium`, `Message`, et `Sujet`, et les autres rôles pouvant être remplis (*Non-Core Frame Elements*) sont le `Destinataire`, la `Quantité_d'information`, un `Descriptif` de l'état de la communication, la `Durée`, la `Fréquence`, la `Façon`, le `Moyen`, l'`Endroit`, le `But`, et le `Moment` correspondant à la communication en question.

Les différents cadres sémantiques sont reliés par des relations de cadre à cadre (par exemple, un cadre peut être un sous-cas d'un autre plus général), pour former une hiérarchie. Un lien de cadre à cadre inclut le plus souvent une correspondance

3. `https://framenet2.icsi.berkeley.edu/fnReports/data/frameIndex.xml?frame=Communication`

entre rôles remplis. Par exemple, le cadre `Engagement` utilise le cadre `Communication` et les rôles `Destinataire`, `Medium`, `Sujet`, et `Manière` sont communs et se correspondent mutuellement entre les deux cadres. Parmi ces rôles, le `Destinataire` est hors noyau pour le cadre de la `Communication`, mais dans le noyau du cadre `Engagement`. La `Manière` est hors noyau pour les deux cadres, et les autres rôles sont dans les noyaux des deux cadres. Le `Communicant` du noyau du cadre `Communication`, n'est pas relié au rôle `Locuteur` du noyau du cadre `Engagement` mais pourrait l'être.

1.2.2 Unités lexicales

Plus de 10 000 unités lexicales sont annotées comme possibles prédicats des différents cadres (à chaque unité lexicale correspond un unique cadre pour lui donner sens). L'unité lexicale comporte une définition, et de nombreux exemples annotés décrivant avec quels rôles du cadre sémantique elle peut se combiner (et dans quels agencements syntaxiques).

Prenons l'exemple de deux unités lexicales, `elbow.v`[4] et `elbow.n`[5] (le verbe *to elbow* et le nom *elbow*).

`Elbow.n` appartient au cadre des `Parties_observables_du_corps` (cadre situationnel plutôt qu'évènementiel), et est défini comme « la jointure entre l'avant-bras et le bras », et est annoté dans 16 phrases. Les rôles du noyau de ce cadre sont la `Partie_du_corps` et le `Possesseur`, et hors noyau, `Attache`, `Descripteur`, `Localisante_orientation`, et `Sous_partie`. Parmi les 16 annotations mentionnant `elbow.n` sont 1 fois annotés les rôles `Attache`, `Descripteur`, et `Localisante_orientation`, et 16 fois les rôles du noyau (mais `Possesseur` est annoté 3 fois comme absent de la phrase, mais compréhensible par le contexte, *Definite Null Instantiation* dans FrameNet). Parmi les 94 unités lexicales qui appellent le même

4. `https://framenet2.icsi.berkeley.edu/fnReports/data/lu/lu13117.xml?mode=lexentry`
5. `https://framenet2.icsi.berkeley.edu/fnReports/data/lu/lu2398.xml?mode=lexentry`

cadre, on trouve aussi `cheville.n`, `barbe.n`, `chevelure.n`, `corne.n`, `gorge.n` et `museau.n`.

`Elbow.v` appartient au cadre `Causer_du_mal`, décrit par : « les mots dans ce cadre décrivent une situation dans laquelle un `Agent` ou une `Cause` blesse une `Victime`. La `Partie_du_corps` de la `Victime` qui est le plus directement affectée peut être mentionnée à la place de la `Victime` ». `Elbow.v` est alors défini comme « frapper ou donner un coup avec le coude (particulièrement dans des compétitions sportives ou pour dégager un chemin) ». Les rôles du noyau de ce cadre sont l'`Agent`, la `Partie_du_corps` touchée, la `Cause` (non-intentionnelle) qui pourrait provoquer le mal, et la `Victime`. Il y a 21 rôles hors noyau, dont le `Degré` de mal causé, les `Circonstances`, les `Itérations` de l'évènement, et le `Résultat` de l'action. Parmi les 13 annotations contenant `elbow.v`, la `Victime` et l'`Agent` sont toujours annotés (l'`Agent` deux fois comme non présent dans la phrase, à cause de sa construction grammaticale, *Constructional Null Instanciation* dans FrameNet), la `Partie_du_corps` 4 fois, et 6 des rôles hors noyau entre 1 et 5 fois. 71 autres unités lexicales appelent ce cadre, dont `cogner.v`, `flageller.v`, `contusionner.v`, `empoisonnement.n`, `empaler.v` et `frappe.n`.

La figure 1.1 montre un exemple d'annotation d'unité lexicale, le nombre de fois où les différents rôles ont été annotés, et leurs réalisations.

1.2.3 Corpus annoté

FrameNet comporte, en termes d'annotations, environ 175 000 exemples (extraits du *British National Corpus*) venant enrichir le dictionnaire d'unités lexicales, et surtout un corpus plus restreint de texte annoté au complet (contrairement aux exemples, ou seul un cadre sémantique est annoté par phrase). Ce corpus d'environ 6000 phrases (pour 25 000 cas de cadres sémantiques) est plus représentatif du texte qu'aurait à traiter une application concrète, et permet d'entrainer et d'évaluer des outils automatiques d'annotation.

Un exemple d'annotation avec `elbow.v` est représenté en figure 1.1.

Frame Element	Number Annotated	Realization(s)
Agent	(13)	CNI.-- (2) NP.Ext (10) PP[by].Dep (1)
Body_part	(4)	PP[in].Dep (4)
Depictive	(2)	PP[with].Dep (1) VPing.Dep (1)
Manner	(1)	AVP.Dep (1)
Purpose	(1)	2nd.-- (1)
Reason	(2)	Sfin.Dep (1) 2nd.-- (1)
Result	(5)	PP[of].Dep (2) VPing.Dep (1) AJP.Dep (1) PP[from].Dep (1)
Time	(1)	PP[after].Dep (1)
Victim	(13)	NP.Ext (3) NP.Obj (10)

I ELBOWED Karen unceremoniously aside and grabbed the paddle .

FIGURE 1.1 – Exemple d'annotation de l'unité lexicale `elbow.v`. *Elbowed* est la cible, *I* remplit le rôle de l'`Agent`, *Karen* celui de la `Victime`, *unceremoniously* décrit la `Manière` et *aside* le `Résultat` de l'événement.

1.3 Analyse sémantique automatique

Étant donné le lourd coût de l'annotation manuelle pour des données aussi précises que celles de FrameNet, on cherche à produire automatiquement ces annotations, et c'est la tâche qui va nous intéresser dans ce mémoire.

Trois systèmes existent et sont librement distribués pour produire automatiquement des annotations suivant FrameNet [6], qui seront abordés à la section 2.

1.3.1 Étapes de l'annotation

L'annotation automatique se déroule généralement en une succession d'étapes :

6. `https://framenet.icsi.berkeley.edu/fndrupal/ASRL`

- **Identifier les prédicats** (ou mots « cibles », *targets* en anglais) qui font référence à des cadres. Typiquement, on peut utiliser la liste des unités lexicales associées à un cadre dans FrameNet (et les cibles annotées dans les données d'entrainement).
- **Désambiguiser le cadre appelé par la cible**, parmi plusieurs cadres similaires et reliés dans la hiérarchie (une unité lexicale n'est listée qu'avec un seul cadre dans FrameNet, mais un mot polysémique correspond à plusieurs unités lexicales).
- **Déterminer la position des arguments** qui remplissent les rôles sémantiques déterminés par le cadre, typiquement à l'aide d'une analyse syntaxique de la phrase.
- **Étiqueter chacun des arguments avec le rôle qu'il remplit.** C'est sur cette sous-tâche (détaillée ci-après) que je me suis concentré.

Les troisième et quatrième étapes peuvent éventuellement être fusionnées, par exemple en recherchant dans la phrase quel segment (ou aucun) remplit un rôle donné – plutôt que de chercher à étiqueter chaque segment – comme il sera détaillé dans la section 2.4.

Typiquement, les prédictions pour chaque argument sont faites indépendamment les unes des autres, et alors une étape de post-traitement peut venir améliorer la cohérence des prédictions entre elles.

1.3.2 Étiquetage de rôles sémantiques

Je me suis concentré sur la tâche qui consiste à assigner son rôle à chaque argument d'un cadre sémantique. Pour chaque occurence d'un cadre dans une phrase, je considère connus :

- la cible ;
- le cadre précis évoqué par la cible ;
- la position de chacun des arguments.

Il s'agit alors de déterminer le bon rôle pour chaque argument parmi la liste de rôles que le cadre accepte (une dizaine en moyenne).

La décision de considérer comme connue la position des arguments est significative, et essentiellement motivée par l'imprécision des analyseurs syntaxiques automatiques. Dans le cas le plus général, un argument peut être composé de n'importe quel sous-ensemble des mots de la phrase, éventuellement disjoints, mais cet espace de recherche est bien trop grand pour trouver les arguments.

Les systèmes d'annotation sémantique s'appuient donc le plus souvent sur une analyse syntaxique de la phrase à annoter, et considèrent comme des arguments potentiels les noeuds de l'arbre syntaxique de la phrase (les linguistes parlent de **constituants**). Malheureusement, les auteurs d'un système à l'état de l'art [7] rapportent qu'environ 20% des arguments annotés ne correspondent pas à un noeud de l'arbre syntaxique [8] produit par l'analyseur qu'ils ont utilisé, et sont dès lors pratiquement impossibles à identifier. Le plus souvent, cela est dû aux erreurs faites par l'analyseur syntaxique. Plus rarement, les arguments annotés dans FrameNet ne sont pas des constituants. Par exemple, dans la phrase *« As capital of Europe's most explosive economy, Dublin seems to be changing before your very eyes. »*, le mot *economy* appelle le cadre `Économie`, et le rôle `Descripteur` de ce cadre est rempli par l'argument *« most explosive »*. Dans une analyse syntaxique de la phrase, le constituant est *« most explosive economy »*, chaque adjectif renvoyant directement à *economy*. L'argument annoté *« most explosive »* n'est pas un constituant.

Lorsque un argument ne correspond pas à un noeud de l'arbre produit par l'analyseur syntaxique, il n'est donc même pas envisagé pour être étiqueté. Qui plus est, comme il sera précisé en section 2.1, la correspondance d'un argument avec un noeud de l'arbre permet de déterminer de nombreuses caractéristiques de cet argument, pertinentes pour son étiquetage. Connaître la position des arguments

7. Das *et al.* (2012)

8. entre autres parce que l'analyseur syntaxique est entrainé sur des données assez différentes de celles annotées dans FrameNet

sans avoir à la prédire évite de souffrir de ces limitations, et simplifie la tâche à traiter.

1.4 Représentations de mots

D'après Turian *et al.* 2010, l'utilisation de représentations distribuées de mots – apprises de manière non-supervisée – est une méthode simple et générale pour améliorer la précision de systèmes d'apprentissage supervisé pour le traitement des langues.

Généralement, dans les systèmes automatiques de traitement des langues, chaque mot est représenté à une place unique d'un lexique, et peut être représenté par un vecteur *one-hot*, c'est-à-dire qui comporte autant de zéros que de mots dans le lexique, avec un seul un à la place du mot représente. Chaque mot du vocabulaire est alors identiquement différent de tous les autres. La représentation des mots par un vecteur de valeurs réelles, équivalent à une position dans un espace, permet de situer des mots nouveaux pour un algorithme d'apprentissage supervisé par rapport à des mots connus, par proximité dans l'espace des représentations. Ainsi les données d'entrainement se généralisent plus facilement à tous les mots dont on a la réprésentation (mais qui sont inconnus du point de vue de l'apprentissage), ce qui améliore les performances globales.

J'ai utilisé les représentations de mots fournies par Ronan Collobert, apprises de manière non-supervisée avec SENNA [9] (Collobert *et al.* 2011), à partir d'un énorme corpus de texte non étiqueté (provenant essentiellement de Wikipédia). Cette ressource fournit la représentation dans un espace de dimension 50 de 130 000 mots, les plus fréquents dans le corpus.

À travers un réseau de neurones (présenté dans Collobert & Weston 2008), à chaque itération de l'algorithme, un n-gram du texte ainsi qu'un n-gram bruité (généré aléatoirement) sont présentés à un modèle de langue, et les représentations

9. `http://ronan.collobert.com/senna/`

des mots de l'historique du n-gram sont actualisées par descente de gradient (par rapport à la différence de score entre le n-gram correct et le n-gram corrompu).

1.5 Mieux généraliser à l'aide de ressources sémantiques

Mon approche vise à utiliser la source de données sémantiques que sont les représentations distribuées des mots du lexique pour améliorer l'apprentissage de l'étiqueteur de rôles sémantiques. Mes données sont donc des correspondances entre les mots (chaines de caractères) et des vecteurs de coordonnées, en dimension 50.

Dans l'idée, si des mots sont proches dans l'espace des représentations, c'est qu'ils sont sémantiquement proches (par exemple, synonymes), et alors l'algorithme d'apprentissage doit se servir de cette proximité pour mieux généraliser sa connaissance des exemples d'entrainement (par exemple, faire de meilleures prédictions pour des exemples de test similaires à des exemples d'entrainement).

1.6 PropBank

PropBank [10] est (par rapport à FrameNet) l'autre ressource principale utilisée pour faire de l'étiquetage sémantique, (voir par exemple Màrquez *et al.* 2008). Le corpus utilisé est celui du Penn Treebank (voir Kingsbury & Palmer 2003), et les annotations de nature sémantique sont rajoutées en tant qu'elles ont un lien avec la syntaxe déjà annotée (le Penn Treebank est une ressource canonique pour la tâche d'annotation syntaxique).

Dans PropBank, les annotations sont focalisées sur les prédicats verbaux (les cibles sont toutes des verbes, et réciproquement tous les verbes sont des cibles), et comme dans FrameNet les arguments peuvent être soit au coeur de l'annotation (*core arguments*), soit plus périphériques (*modifiers*). La plus grande différence

10. `http://verbs.colorado.edu/~mpalmer/projects/ace.html`

concerne le type d'annotation que constituent les arguments essentiels : dans PropBank, les rôles principaux de tous les verbes sont les mêmes, étiquetés de A0 à A5. A0 et A1 sont les rôles les plus fréquents, et annotent respectivement l'`Agent Prototypique` et le `Patient Prototypique` (ou le `Thème`).

Dans FrameNet, chaque cadre sémantique définit les rôles qui lui sont propres, alors que dans PropBank, les rôles sont complètement abstraits des prédicats individuels : cette abstraction peut être intéressante pour un algorithme d'apprentissage si les arguments recouverts par les mêmes étiquettes ont effectivement de forts points communs (et PropBank a été conçu dans cette optique), mais en pratique, il est difficile de savoir ce que généralisent exactement les arguments de A3 à A5.

Dans la mesure où les représentations de mots que j'emploie sont assez précises lexicalement, et sans rapport avec la syntaxe, il est intéressant de chercher à annoter des rôles sémantiques précis plutôt que très généraux, et de suivre les conventions de FrameNet plutôt que de PropBank. C'est le choix que j'ai fait dans ce mémoire.

CHAPITRE 2

ÉTAT DE L'ART

L'annotation automatique de rôles sémantiques a été largement étudiée depuis l'article fondateur de Daniel Gildea et Daniel Jurafsky (2000). Le développement de PropBank (Kingsbury & Palmer 2002, Palmer *et al.* 2005), et les tâches CoNLL de 2004 et 2005 (Carreras & Màrques 2004, Carreras & Màrquez 2005) ont beaucoup influencé l'annotation sémantique (voir par exemple Koomen *et al.* 2005, Punyakanok *et al.* 2008). Màrquez *et al.* (2008) donne une bonne vue d'ensemble de la tâche, des influences de la linguistique sur les ressources existantes, ainsi que des techniques et des heuristiques courantes. Des travaux ont également été menés pour d'autres langues que l'anglais, comme Coppola & Moschitti (2010).

Pour FrameNet spécifiquement, plusieurs auteurs se sont attachés à identifier les arguments et leurs rôles, à commencer par Gildea & Jurafsky (2000) (section 2.1). Fleischman *et al.* (2003) ont été les premiers à utiliser des modèles à maximum d'entropie pour cette tâche (section 2.2). Des systèmes complets ont aussi été développés pour identifier automatiquement la totalité de l'annotation sémantique (le prédicat, le cadre, les arguments et leurs rôles), notamment avec la tâche 19 de la campagne d'évaluation SemEval-2007 (Baker *et al.* 2007). SEMAFOR [1] est un tel système, développé depuis 2010 à l'université Carnegie-Mellon, et constitue l'état de l'art en la matière (voir section 2.4). Les autres systèmes distribués librement sont LTH [2] (auquel se compare SEMAFOR), développé à l'université de Lund, en Suède (Johansson & Nugues 2007), et Shalmaneser [3], de l'université de la Sarre, en Allemagne (Erk & Pado 2006).

1. `http://www.ark.cs.cmu.edu/SEMAFOR/`
2. `http://nlp.cs.lth.se/software/semantic_parsing_framenet_frames/`
3. `http://www.coli.uni-saarland.de/projects/salsa/shal/`

2.1 Gildea & Jurafsky (2000)

Gildea & Jurafsky (2000) présente le premier système d'identification des rôles sémantiques de constituants d'une phrase dans un certain cadre sémantique, en utilisant des caractéristiques lexicales et syntaxiques extraites d'une analyse automatique de la syntaxe de la phrase. Ils décomposent la tâche en deux parties : la détection des frontières des arguments, marginalement, et surtout la classification de leur rôle.

Les caractéristiques employées sont le type syntaxique de l'argument (nominal, verbal, etc.), sa fonction grammaticale dans la phrase, sa position par rapport au prédicat, la voix (active ou passive) du verbe, et le mot de tête du syntagme. Toutes ces notions étant définies pour des constituants de la phrase (des noeuds de l'arbre syntaxique), il est primordial que les arguments à annoter correspondent à des syntagmes identifiés comme tels par l'analyseur syntaxique. S'il n'y a pas de correpondance exacte, il s'agit de déterminer heuristiquement parmi les noeuds de l'analyse syntaxique celui qui couvre le mieux l'argument.

Pour estimer les probabilités des rôles, les auteurs utilisent une cascade de modèles (*backoff*) avec des couvertures variables, les plus précis employant le plus de caractéristiques et couvrant le moins d'exemples. Lorsque toutes les caractéristiques ne sont pas utilisables (pour des valeurs non vues à l'entrainement), les modèles utilisant moins de caractéristiques, moins précis mais avec une plus grande couverture, sont utilisés. Par exemple, le modèle qui utilise le mot de tête, le type de syntagme, et la cible, peut se prononcer dans 50% des cas, et a une précision de 87%, alors que le modèle n'utilisant que la cible couvre 100% des cas mais est seulement précis à 41%.

Le modèle avec le meilleur modèle de rappel, réglé sur leur ensemble de développement, obtient 76.9% de précision sur leur ensemble de test. Leurs données étaient les phrases exemples d'une version préliminaire de FrameNet comportant 67 cadres, soit 49 013 phrases annotées avec un cadre par phrase.

Les auteurs observent que les mots de tête constituent une caractéristique des plus fiables pour déterminer le rôle d'un argument : le modèle utilisant uniquement la cible et le mot de tête pour classifier un rôle obtient 86,7% de précision, mais ne couvre que 56% des données. Pour augmenter la couverture de ce modèle, les auteurs font une expérience pour grouper les noms du lexique (avec la technique de *clustering* décrite dans Lin (1998)), ce qui leur permet d'obtenir un modèle couvrant 98% des mots de tête nominaux, et précis à 79.7%. Finalement, cela ajoute 0,8% de précision à leur modèle global, sur l'ensemble de développement.

Mon étude vise à être la continuation de cette expérience, en utilisant les représentations distribuées de mots pour généraliser les mots du lexique.

2.2 Fleischman & Hovy (2003)

En 2003, Fleischman et Hovy ont pour la première fois utilisé des modèles à maximum d'entropie pour classifier les rôles d'arguments de cadres sémantiques, en obtenant des résultats proches de ceux de Gildea et Jurafsky (76% de précision), sur des données comparables (40 000 phrases exemples de FrameNet).

Les auteurs apportent une amélioration notable au modèle de base : ils cherchent à classifier les différents arguments d'une même phrase en tant qu'ils forment une séquence, plutôt qu'individuellement. Pour cela, les étiquettes des deux précédents arguments sont rajoutés aux caractéristiques de l'argument à étiqueter à l'entrainement des modèles, et l'algorithme de Viterbi est utilisé pour étiqueter les séquences d'arguments de l'ensemble de test. Cet étiquetage joint leur permet d'améliorer leurs performances d'environ 1,5%.

Fleischman et Hovy comparent enfin l'utilisation de caractéristiques extraites automatiquement de l'analyse syntaxique de la phrase à annoter, et de données annotées manuellement (fournies par FrameNet). Les annotations manuelles améliorent nettement la performance des classifieurs, d'environ 9% (pour atteindre environ 85% de précision). De telles données sont très couteuses à obtenir, et ne

reflètent pas les possibilités d'annoter un autre texte, mais démontrent les progrès qu'une meilleure analyse syntaxique automatique permettrait d'obtenir.

2.3 Shalmaneser : Erk & Pado (2006)

Élaboré en 2006 , Shalmaneser [13] est le premier système complet d'analyse sémantique avec FrameNet distribué librement, et conçu comme un outil unifiant des modules indépendants procédant à l'analyse (avec des étapes semblables à celles mentionnées section 1.3.1).

Les modules développés pour Shalmaneser utilisent des classifieurs de Bayes naïfs et des modèles à maximum d'entropie, les auteurs mettant particulièrement l'accent sur la souplesse de l'outil, compatible avec plusieurs modules d'apprentissage machine externes. L'utilisateur a le choix de prédire les rôles des arguments en une ou deux étapes (autrement dit, une étape de classification entre « argument » et « pas un argument » est optionnelle). De nombreuses autres fonctionnalités relatives à l'indépendance des modules sont offertes par cet outil (comme l'utilisation d'analyseurs syntaxiques différents, ou encore l'entrainement de modèles pour différents sous-ensembles des données, cadre par cadre ou cible par cible, etc.).

Sur les exemples de la version 1.2 de FrameNet, les auteurs rapportent une précision de 78% pour l'étiquetage des rôles des arguments (la tâche *arglab* du module ROSY), grâce à des caractéristiques syntaxiques et lexicales.

2.4 SEMAFOR : Das *et al.* (2010)

SEMAFOR est un système complet automatique d'annotation sémantique, développé depuis 2010 à l'université Carnegie-Mellon. Comme élaboré en section 1.3.1, le système identifie tour à tour :

– les cibles, grâce à des règles sur les lemmes des cibles vues dans l'ensemble d'entrainement ;

– les cadres auxquels réfèrent les cibles, lorsqu'elles sont ambiguës, grâce à un modèle log-linéaire basé sur les dépendances syntaxiques de la cible, et sur son lemme [4] ;
– les arguments avec leur rôle.

Plutôt que d'identifier d'abord les arguments pour ensuite leur assigner un rôle – comme il est courant dans la littérature sur l'étiquetage sémantique, par exemple dans Koomen *et al.* (2005), Xue & Palmer (2004), ou Johansson & Nugues (2007) – SEMAFOR identifie, pour chaque rôle, quel segment de phrase le remplit le mieux (ou le segment vide, quand le rôle n'est pas exprimé dans la phrase). Les étapes d'identification et de classification des arguments sont donc menées simultanément.

Le modèle employé pour la classification des rôles des arguments est un deuxième modèle log-linéaire, basé sur environ un million de caractéristiques binaires des arguments (caractéristiques lourdement lexicalisées sur les dépendances syntaxiques, les parties du discours, et les lemmes de plusieurs mots dans l'arguments, ainsi que sur la relation entre l'argument et la cible).

Par comparaison avec ma tâche, étant données la cible, le cadre, et les positions exactes des arguments, SEMAFOR obtient un score F_1 [5] de 80,5% avec ce modèle pour étiqueter indépendamment le rôle des arguments, sur la version 1.3 de FrameNet.

SEMAFOR inclut aussi une étape de post-traitement des prédictions faites indépendamment les unes des autres, qui vise principalement à empêcher que les arguments prédits pour remplir différents rôles ne se superposent. Ce procédé améliore leur score F_1 sur la tâche qui nous intéresse de 0,5%, pour atteindre 81%.

4. Voir aussi Das & Smith 2011 pour l'apprentissage semi-supervisé de cadres appelés par des cibles non vues à l'entrainement.

5. Avec son approche de détection des arguments remplissant chaque rôle, connaissant la position de l'ensemble des arguments, SEMAFOR doit encore déterminer quels rôles sont exprimés et lesquels ne le sont pas. Ses prédictions sont assez précises (88%), mais avec un plus faible rappel (75%).

2.5 Apprentissage joint des différents cadres sémantiques

Une des particularités notables de SEMAFOR, en particulier par rapport à LTH duquel il s'inspire et auquel il se compare, est que les deux modèles de désambiguisation des cadres, et surtout de classification des rôles sont communs à tous les cadres sémantiques, plutôt que d'être entrainés séparément. LTH utilise de nombreux séparateurs à vaste marge : pour désambiguiser les cadres, pour identifier les arguments, et pour classifier les arguments – avec un modèle par cadre sémantique.

N'étant pas persuadé que l'apprentissage des modèles des différents cadres puisse s'unifier raisonnablement (autrement dit, qu'une connaissance de l'apprentissage des autres cadres serve pour apprendre le modèle d'un cadre particulier), j'ai préféré entrainer un modèle par cadre comme il est plus courant (et plus simple). L'idée serait à creuser davantage, en particulier pour des cadres sémantiques proches. Les relations de cadre à cadre permettraient de distribuer les exemples annotés entre plusieurs cadres pour en agrandir l'ensemble d'entrainement, notamment. Plusieurs cadres partagent aussi des rôles similaires, même si les situations sont très différentes, et ces exemples aussi pourraient être généralisés.

Il me semble toutefois qu'une telle exploration demanderait davantage d'effort manuel ou semi-automatique pour faire la part des connaissances généralisables et des distributions de rôles entièrement indépendantes, ce qu'un modèle log-linéaire ou à maximum d'entropie (équivalents à un réseau de neurones à une seule couche) sont incapables de détecter. Cet effort sort du cadre de ce mémoire.

CHAPITRE 3

EXPÉRIENCES ET RÉSULTATS

Dans mes expériences, étant donné un cadre sémantique appelé par une cible connue, et étant donnés des arguments, un classifieur doit étiqueter chaque argument avec le bon rôle, parmi ceux définis pour le cadre sémantique par FrameNet. Du point de vue de l'apprentissage machine, c'est un problème bien connu de classification multi-classe (voir par exemple Kotsiantis 2007).

3.1 Données

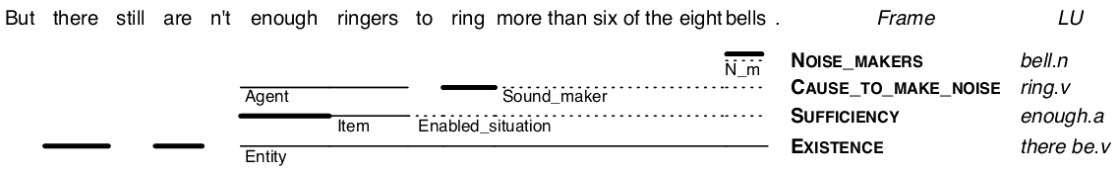

FIGURE 3.1 – Exemple d'annotation de tous les cadres présents. Le cadre `Cause_de_bruit` (*Cause to make noise*) est appelé par l'unité lexicale *ring.v*, la cible. Dans ce cadre, les rôles `Agent` et `Producteur_de_son` sont annotés, remplis par *enough ringers* et *more than six of the eight bells* respectivement. Les cadres `Faiseur_de_bruit`, `Suffisance` et `Existence` sont également annotés dans la même phrase.

Les données complètement annotées de FrameNet représentent 5946 phrases, réparties parmi 78 documents. J'ai utilisé le même découpage des données que (Das & Smith 2011), avec 55 documents pour l'ensemble d'entrainement et 23 documents pour le jeu de test. L'ensemble d'entrainement comporte aussi 123 190 phrases exemples (annotées avec un cadre par phrase).

Sur 1019 cadres définis par FrameNet, 470 ont des exemples dans l'ensemble de test, et peuvent être évalués. Comme les différents cadres sémantiques sont largement indépendants les uns des autres, un modèle est entrainé par cadre.

L'ensemble d'entrainement (restreint aux 470 cadres testables) se compose de 142 659 instances de cadres sémantiques, avec en moyenne 1,08 argument annoté par cadre (soit 154 769 arguments annotés). En moyenne, le modèle d'un cadre testé a 325 exemples d'entrainement.

Dans FrameNet, en moyenne 10 rôles sont définis pour un cadre donné, et dans les données d'entrainement, on rencontre en moyenne 4,7 rôles différents annotés au moins une fois. Les différents rôles sont assez bien représentés dans les données : un modèle simpliste qui annoterait pour tous les arguments d'un cadre le rôle le plus fréquemment vu à l'entrainement pour ce cadre obtient 50% de précision.

3.2 Évaluation

L'ensemble de test comporte 4456 instances de cadres, soit 7209 arguments à classifier. Pour permettre une certaine comparaison à l'état de l'art [1], la performance rapportée pour chacune des méthodes évaluées est la micro-précision sur cet ensemble d'arguments, c'est-à-dire la proportion d'arguments pour lesquels le rôle effectivement est prédit par le modèle, tous cadres confondus.

Le système fait une prédiction pour chaque argument à annoter, aussi le rappel et la F_1-mesure sont tous deux égaux à la précision rapportée.

3.3 Système de référence

3.3.1 Présentation

Comme système de référence, et dans la lignée de Das *et al.* 2010 et Fleischman & Hovy 2003, j'ai entrainé – pour chaque cadre, contrairement à SEMAFOR, cf section 2.5 – un modèle à maximum d'entropie (équivalent à une régression logistique). Le modèle s'appuie sur plusieurs caractéristiques (*features*) de l'argument

1. à nuancer par les différents ensembles de données utilisés par les différents auteurs

à classifier, portant sur sa position et sur son contenu, et prédit le rôle le plus probable compte tenu de ces données.

Le modèle distribue la probabilité des différentes classes pour maximiser la vraisemblance des connaissances de l'ensemble d'entrainement, autrement dit pour faire les prédictions les moins arbitraires possibles (le maximum d'entropie au sens de Shanon) pour les exemples de test. Voir par exemple Ratnaparkhi (1997) pour une présentation de l'usage typique de ces modèles en traitement des langues.

Le modèle à maximum d'entropie a été implémenté avec NLTK [2] (Wagner 2010), bibliothèque pour le traitement des langues écrite en Python.

3.3.2 Caractéristiques employées

Les caractéristiques du modèle de référence se décomposent en plusieurs catégories. Avec l'exemple de l'`Agent` dans le cadre `Cause_de_bruit` (figure 3.1), les caractéristiques employées sont présentées dans la table 3.I.

Les prédictions de rôle étant faites individuellement pour chaque argument, la dernière caractéristique permet au modèle de tenir compte dans une certaine mesure du cadre dans lequel a lieu chaque prédiction. Intuitivement, dans les cadres où de tels rôles s'instancient, un argument qui est le premier annoté dans la phrase a peu de chances d'être le Patient ou le Thème de l'action, et plus de chances d'être l'Agent (au sens du cadre spécifique en question).

Les parties du discours (*part of speech*) ont été étiquetées automatiquement à l'aide du Stanford Parser (de Marneffe *et al.* 2006), en même temps qu'ont été obtenues les dépendances syntaxiques de la phrase.

3.3.3 Résultats

On peut voir les performances des différents groupes de caractéristiques (utilisées séparément) en table 3.II.

2. `http://nltk.org/`

18. de plus de 5 caractères

TABLE 3.I – Caractéristiques du modèle de référence, rangées par catégorie, ainsi que leur valeur dans l'exemple présenté en figure 3.1.

Caractéristiques par catégorie	Valeur dans l'exemple
Caractéristiques de base	
le texte de l'argument	`enough ringers`
le texte de la cible	`ring`
la position (en caractères) de l'argument dans la phrase	23
Position relative	
est-ce que l'argument est avant ou après la cible	avant
si l'argument est à la même place que la cible	non
la distance (en mots) entre la cible et l'argument	1
Nombre de mots	
le nombre de mots de l'argument	2
le nombre de mots pleins [3]de l'argument	2
Contenu	
le premier mot de l'argument	`enough`
le premier mot plein de l'argument	`enough`
Parties du discours	
partie du discours du premier mot de l'argument	JJ (adjectif)
partie du discours du dernier mot de l'argument	NNS (nom pluriel)
la partie du discours majoritaire dans l'argument	JJ
Arguments précédents de cadre	
nombre d'arguments déjà étiquteés dans le cadre	0

La table 3.III montre les meilleures performances en utilisant seulement n groupes de caractéristiques, avec n entre 1 et 6, parmi les groupes évalués en table 3.II. Essentiellement, davantage de caractéristiques amènent de meilleurs résultats, mais comme on peut voir dans les dernières lignes, pas absolument toujours.

Pour les expériences utilisant les représentations de mots présentées dans la suite, tous les groupes de caractéristiques sont employés, car toutes les caractéristiques améliorent (un peu, comme on verra) les performances du modèle.

Finalement, ce système de référence arrive à 80.0% de précision, à comparer aux 76% de précision (avec des données obtenues automatiquement) de Fleischman & Hovy (2003), ou aux 80.97% de F_1-mesure de Das *et al.* (2010), sur d'autres versions des données de FrameNet. L'utilisation de représentations de mots va venir améliorer ce système de référence.

TABLE 3.II – Performances des caractéristiques du modèle de référence, classées par groupe.

Catégories	Précision
Base	60.5%
Position relative	73.8%
Nombre de mots	57.7%
Contenu	58.9%
Parties du discours	66.5%
Ordre des arguments	61.1%

TABLE 3.III – Performances des combinaisons de groupes de caractéristiques du modèle de référence.

Nombre et catégories utilisées	Précision
1 : Position relative	73.8%
1 : Parties du discours	66.5%
2 : Position relative + Contenu	77.3%
2 : Position relative + Parties du discours	77.6%
3 : Base + Position relative + Parties du discours	78.7%
3 : Position relative + Ordre + Parties du discours	77.8%
4 : Base + Position relative + Contenu + Parties du discours	79.4%
4 : Position relative + Ordre + Contenu + Parties du discours	79.4%
5 : Base + Position relative + Ordre + Contenu + Parties du discours	80.0%
6 : Tous les groupes	79.6%

3.4 Utilisation des représentations de mots

3.4.1 Représentation des arguments

50% des arguments sont constitués de plus d'un mot. Pour pouvoir déterminer des caractéristiques de ces arguments à partir des représentations de mots, il faut un moyen de sélectionner un ou plusieurs mots de leur texte qui seraient les plus significatifs. Je suis parti du principe qu'un mot appartenant à chaque argument devait le représenter sémantiquement : idéalement, *l'ambassadeur iraquien* est représenté par *ambassadeur*, proche de *personne*, ou de *porte-parole* parmi les mots vus à l'entrainement, et ces derniers remplissent fréquemment le même rôle (`Locuteur`) dans le cadre de l'`Engagement`, ce qui permettrait de conclure que *l'ambassadeur iraquien* remplit aussi le rôle de `Locuteur`.

Déterminer quel mot représente le mieux un argument est une tâche en soi. J'ai testé 3 méthodes simples (le premier mot, le dernier mot, ou le mot le plus long), et une méthode s'appuyant sur les dépendances grammaticales [4] de la phrase : le mot de tête d'un segment de phrase est le mot situé le plus haut dans la hiérarchie des dépendances.

Par exemple, l'argument *their ignorance which was based on prominent views* est représenté par *ignorance*, car toute la proposition subordonnée dépend (indirectement) de *based*, qui dépend de *ignorance*, et *their* dépend aussi d'*ignorance*. Dans le cadre du `Jugement` (fait par une personne, pouvant être positif ou négatif), cet argument est ensuite classifié comme `Celui_ou_ce_qui_est_jugé`.

Les performances des différentes méthodes sont rapportées en table 3.IV en ajoutant leur mot représentatif aux caractéristiques des arguments. Le modèle du plus proche voisin (détaillé ci-après) donne les mêmes résultats de façon beaucoup plus marquée (voir table 3.V). Globalement, les mots de tête sont les plus pertinents pour représenter sémantiquement les arguments.

TABLE 3.IV – Précision du modèle de référence enrichi de la connaissance du mot représentatif des arguments, en fonction du mot choisi pour les représenter. Les mots de tête sont déterminés grâce aux dépendances obtenues avec le Stanford Parser. L'utilisation des mots de tête donne systématiquement de légèrement meilleurs résultats.

Mot représentatif	Précision
Premier mot	80.3%
Mot de tête	80.4%
Mot le plus long	80.2%
Dernier mot	80.3%

Une fois un argument représenté par un mot, on peut lui assigner une position dans l'espace des représentations, celle du mot qui le représente. 2% des mots de tête des arguments n'ont pas de représentation dans les données employées [5]. Pour ceux-là, seules les caractéristiques de base sont employées.

4. `http://nlp.stanford.edu/downloads/stanford-dependencies.shtml`
5. Tous les nombres sont représentés par le mot `number`.

3.4.2 Plus proche voisin

Dès lors que chaque argument a une position dans l'espace des représentations, l'algorithme de classification qui vient naturellement à l'esprit est celui des plus proches voisins (Cover & Hart 1967) : en phase de test, un argument à classifier a une position dans l'espace des représentations, les différents exemples d'entrainement aussi, et on peut calculer des distances entre ces positions. On peut alors assigner à un argument l'étiquette la plus fréquente parmi ses plus proches voisins. Pour reprendre l'exemple, le plus proche voisin de l'argument représenté par *ambassadeur* serait celui représenté par *porte-parole*, donc on lui assigne le même rôle, `Locuteur`.

Le modèle du 1-plus-proche-voisin – qui prédit pour un argument le rôle de l'exemple d'entrainement le plus proche – obtient 70% de précision. C'est assez remarquable pour un modèle aussi simple. Pour comparaison, le modèle qui prédit pour un argument le rôle le plus fréquemment annoté à l'entrainement arrive seulement à 50% de précision.

Le modèle à maximum d'entropie est généralement plus puissant que celui des plus proches voisins. Concrètement, en entrainant les modèles des cadres avec comme seule caractéristique la prédiction du modèle des plus proches voisins, on obtient un meilleur résultat, détaillé en table 3.V, en fonction du choix de mot représentatif.

TABLE 3.V – Précision du (1) plus proche voisin en fonction du mot choisi pour représenter chaque argument. L'utilisation des mots de tête donne encore de meilleurs résultats.

Mot représentatif	Précision
Premier mot	73.8%
Mot de tête	74.0%
Mot le plus long	68.4%
Dernier mot	68.9%

3.4.3 *k* plus proches voisins

Prendre en compte plusieurs voisins dans la prédiction du rôle d'un argument réduit le bruit dans les données et améliore nettement les performances. Toutefois, prendre en compte trop de voisins diminue l'importance de la proximité entre les mots (à l'extrême, en considérant tous les voisins, leur position n'a plus d'importance). On peut pondérer le rôle de chaque voisin en fontion de la distance, ce que j'ai essayé de faire (voir section 3.4.4). La figure 3.2 montre les résultats en fonction de k du modèle des plus proches voisins (en réalité, un modèle à maximum d'entropie ayant pour seule caractéristique cette information pour classifier chaque argument).

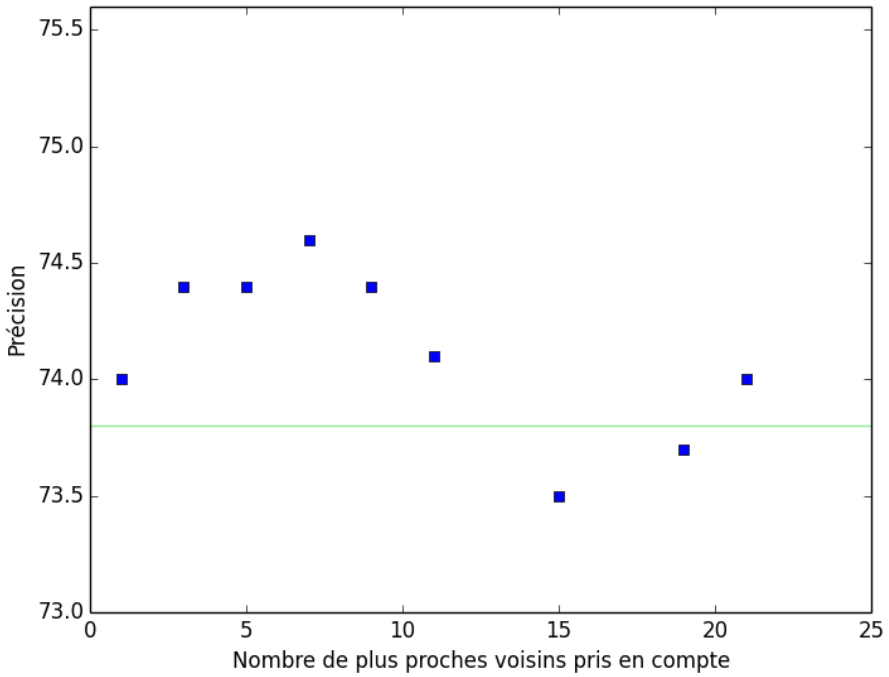

FIGURE 3.2 – Précision moyenne du modèle utilisant pour seule caractéristique le rôle majoritaire parmi les k plus proches voisins de chaque argument. La ligne représente la meilleure caractéristique du modèle de référence, considérées séparément.

Pour intégrer cette information au modèle de référence, on peut simplement

ajouter aux caractéristiques d'un argument le rôle prédit par le modèle des plus proches voisins. La figure 3.3 montre les performances obtenues : avec plusieurs voisins, cette information améliore le modèle de référence (en vert), et permet d'arriver au niveau du système SEMAFOR, à l'état de l'art (en rouge pointillé).

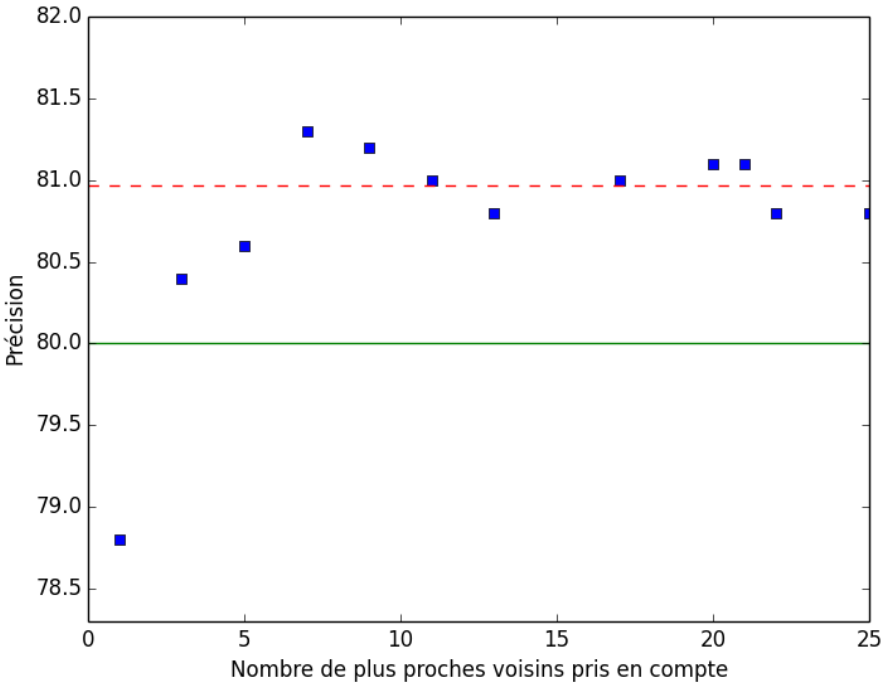

FIGURE 3.3 – Précision du modèle de référence informé de la prédiction du modèle des k plus proches voisins. La ligne verte représente le système de référence, et la ligne rouge pointillée SEMAFOR.

3.4.4 Voisins pondérés

J'ai essayé de raffiner la méthode des plus proches voisins en les pondérant de différentes façons. D'après (Tan 2005), lorsque différentes classes sont réparties inégalement dans les données d'entrainement, l'algorithme des plus proches voisins identifie la classe la plus fréquente avec une grande précision, mais au détriment des classes les moins fréquentes.

Pour remédier à ce phénomène, les exemples des classes les moins fréquentes peuvent être surpondérés pour compenser leur rareté. Dans une expérience, j'ai pondéré chaque exemple par l'inverse de la fréquence de sa classe. La figure 3.4 montre les résultats en fonction du nombre de voisins pris en compte. Ça n'est pas efficace : les rôles plus rares sont peut-être mieux repérés, mais au détriment des rôles plus fréquents.

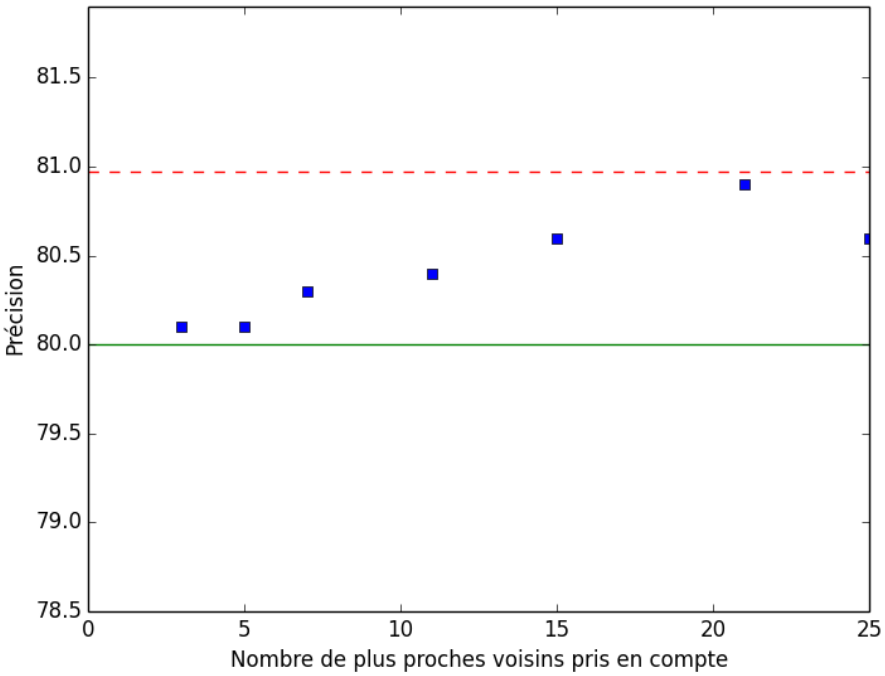

FIGURE 3.4 – Référence + Modèle des k plus proches voisins pondérés par l'inverse de leur fréquence, pour mieux étiqueter les rôles rares.

Une autre façon classique de pondérer les plus proches voisins est par l'inverse de la distance, ce que j'ai fait dans l'expérience suivante (figure 3.5). C'est efficace, et indifféremment du nombre de voisins considérés (entre 3 et 15), les performances dépassent 81%.

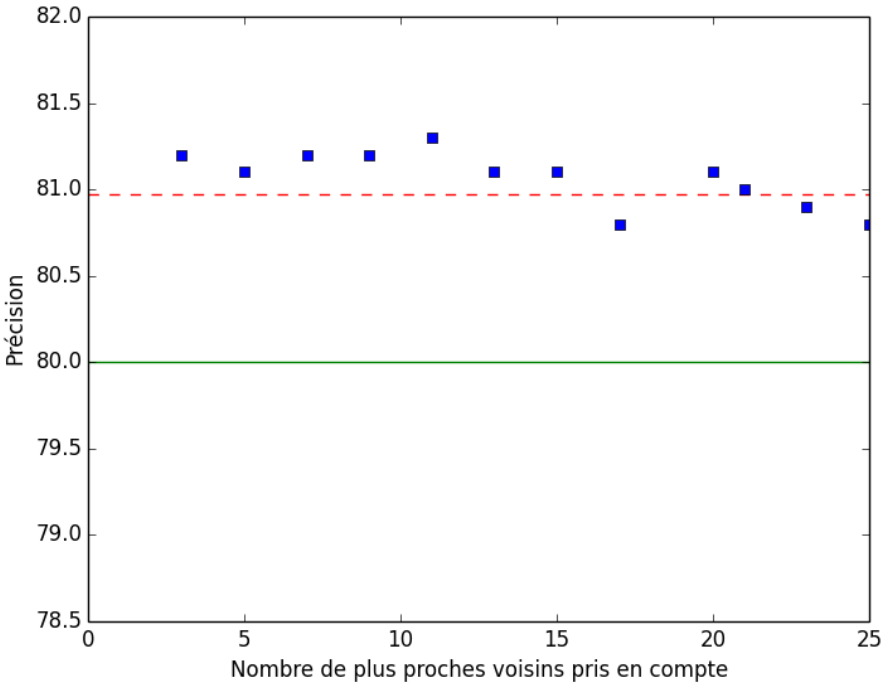

FIGURE 3.5 – Référence + Modèle des k plus proches voisins pondérés par l'inverse de leur distance.

3.4.5 Centres des rôles

Une autre façon d'assigner un rôle à un mot inconnu consiste à trouver de quel rôle il est le plus probablement un représentant : dans l'espace des représentations, on peut situer la position moyenne des mots représentant un rôle, et assigner à un mot le rôle du centre dont il est le plus proche. Cela revient à partitionner l'espace en grandes cellules de Voronoï, une par rôle, centrées aux positions moyennes des exemples d'entrainement des classes, et à classifier les arguments suivant ces cellules. Seule, cette caractéristique obtient 73.4% de précision. En addition au modèle de référence, elle permet d'atteindre 81.1%.

On peut enfin combiner ce modèle avec celui des plus proches voisins (sans pondération), et ajouter les deux prédictions au modèle de référence. La figure 3.6 montre les résultats d'une telle combinaison. La meilleure performance est de

81.5%, avec 20 voisins.

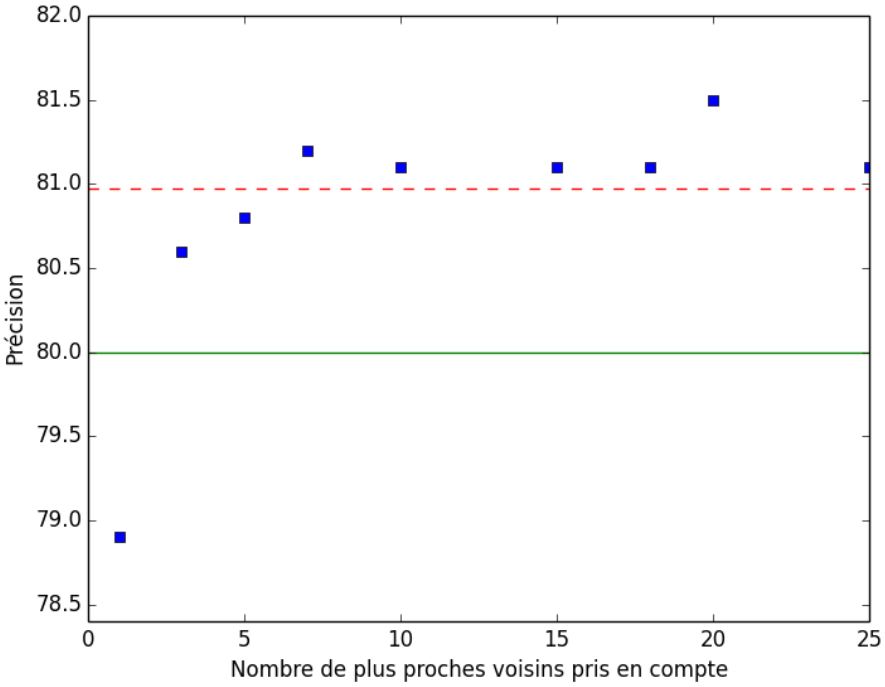

FIGURE 3.6 – Précision des modèles combinés de reférence, des plus proches voisins et des centres des rôles.

On peut essayer de limiter les exemples d'entrainement pris en compte pour calculer les centres des rôles, mais après quelques expériences, cela parait être moins bon que de considérer tous les exemples.

3.5 Analyse

La table 3.VI récapitule les différents résultats.

En regardant les résultats en détail, on observe que les gains en performance proviennent en grande partie des cadres avec le moins d'exemples. La figure 3.VII montre la moyenne des précisions des modèles de chaque cadre (c'est-à-dire non pondérée par le nombre d'exemples de test que le modèle a à classifier), pour les

TABLE 3.VI – Récapitulatif des performances – la performance d'une méthode est la proportion d'arguments correctement étiquetés dans l'ensemble de test, tous cadres confondus (aussi appelée micro-précision).

Modèle	Performance
Centres des rôles	73.4%
k plus proches voisins	74.6%
Fleischman et Hovy (2003)	76%
Référence	80.0%
Référence + mot représentatif	80.4%
SEMAFOR (2010)	81.0%
Référence + centres	81.1%
Référence + plus proches voisins	81.3%
Référence + voisins pondérés	81.3%
Référence + centres + PPV	81.5%

différentes méthodes. On peut voir que l'apport des représentations de mots est plus net avec cette mesure. En particulier, les 100 cadres sémantiques (environ 20% des cadres) avec le moins d'exemples d'entrainement améliorent leur précision moyenne de 5 points lorsque l'on rajoute la prédiction utilisant les centres des rôles au modèle de référence.

Cette observation est cohérente avec la vision évoquée en section 2.1, à savoir que l'abstraction sur des mots du lexique que permettent les représentations de mots permet de mieux généraliser les données disponibles, surtout lorsqu'elles sont peu importantes (peu d'exemples par classe). Cette mesure est d'autant plus pertinente que les cadres peu représentés dans l'ensemble de test sont aussi les cadres les plus difficiles à entrainer, du fait de leur plus faible nombre d'exemples d'entrainement.

On peut aussi remarquer que lorsque les représentations de mots sont utilisées, la prédiction du modèle basé sur elles est considérée comme importante par le modèle à maximum d'entropie, c'est-à-dire qu'il lui alloue un poids important. Par exemple, avec les centres des rôles, le rôle du centre le plus proche est la caractéristique la plus lourdement pondérée dans 77% des modèles des cadres [6] ; pour comparaison, avec

6. la deuxième caractéristique la plus pondérée l'est dans seulement 7% des cadres

TABLE 3.VII – Performance calculée en faisant la moyenne des précisions des modèles de chacun des cadres sémantiques (macro-précision). Les cadres avec moins d'exemples dans l'ensemble de test prennent davantage d'importance, par rapport au calcul de la micro-précision.

Modèle	Précision moyenne des cadres
Centre des rôles le plus proche	69.6%
k plus proches voisins	70.0%
Référence	73.2%
Référence + mot représentatif	74.0%
Référence + centres	75.8%
Référence + plus proches voisins	75.2%
Référence + voisins pondérés	74.8%
Référence + centres + PPV	76.1%

le modèle de référence, la caractéristique la plus pondérée (est-ce que l'argument est avant ou après la cible) l'est dans seulement 30% des modèles des cadres.

En bilan, on peut dire que les représentations de mots apportent de la robustesse au modèle. Une approche simple (k plus proches voisins) donne à elle seule des performances raisonnables (75% de précision), et elles permettent aussi d'enrichir et d'améliorer les performances d'un modèle semblable à l'état de l'art.

Les erreurs restantes sont en partie dues aux arguments prépositionnels (dont le mot de tête est une préposition). Avec la liste de mots vides (*stopwords*) de NLTK, les arguments dont le mot représentatif est un mot vide représentent 40% des erreurs, pour seulement 27% des données : le taux d'erreur pour ces arguments est de 26%, contre 19% en général. Surdeanu *et al.* (2003) propose de généraliser le concept de mot de tête pour en faire un « mot de contenu », et propose de nouvelles règles heuristiques pour déterminer le mot de tête dans le cas de syntagmes prépositionnels. Ce « mot de contenu » est la direction à poursuivre pour étendre l'utilisation adéquate des représentations de mots à tous les arguments.

3.6 Étiquetage joint des arguments d'un cadre

Jusqu'à présent, à part la caractéristique « nombre d'arguments précédents », les rôles des différents arguments d'un même cadre sont prédits de façon indépendante. Or, les annotations sont assez dépendantes les unes des autres. Par exemple, deux arguments d'une phrase ont le même rôle dans seulement 5% des cas (et seulement 20% des cadres le permettent), alors que le modèle peut assigner indifféremment à plusieurs arguments le même rôle s'il l'estime le plus probable pour chacun (en pratique, cela arrive dans 6% des cas, pour les cadres où une telle répétition est impossible). J'ai essayé de lutter contre ce phénomène, en corrigeant *a posteriori* les décisions prises pour éviter les répétitions de rôles dans les cadres où c'est indésirable.

Lorsque le même rôle est prédit pour plusieurs arguments d'une même phrase, le modèle choisit une autre séquence de prédictions sans répétition pour les différents arguments, qui maximise la probabilité totale des décisions individuelles. Grâce au faible nombre d'arguments à annoter par cadre, et au faible nombre de rôles possible, il est facile de parcourir l'ensemble des combinaisons possibles dans la plupart des cas[7]. Quand il y a trop de combinaisons possibles (environ 15% des cas), le modèle garde la séquence de prédictions avec une répétition.

Malheureusement, si une telle correction permet bien de diminuer les prédictions incompatibles (deux arguments avec le même rôle dans un cadre qui ne le permet pas), la nouvelle prédiction n'est pas meilleure pour autant, et les performances se dégradent légèrement (de 0.5% en moyenne).

Dans le même ordre d'idées, on peut trouver dans FrameNet différentes informations supplémentaires. Dans certains cadres, des rôles sont mutuellement exclusifs : par exemple, dans le cadre du `Bruit_de_déplacement`, un argument peut avoir le rôle de `Zone` (dans laquelle a lieu le déplacement) ou bien un argument peut

7. Dans 75% des cas, il y a moins de 500 combinaisons possibles, et dans 87% des cas, moins d'un million.

avoir le rôle de `Chemin` (suivant lequel a lieu le déplacement), mais les deux sont incompatibles. Au contraire, des rôles peuvent aller de pair et la présence de l'un requiert la présence de l'autre. Pour d'autres rôles, il est annoté un type sémantique (parmi une petite ontologie propre à FrameNet), auquel doit appartenir le référent de l'argument (par exemple, `Sentient`, `Message`, `Lieu`, `Degré`, etc.). Malheureusement, chacune de ces informations ne concerne qu'une petite fraction des rôles ou des cadres (10% des rôles ont un type sémantique), et il faut consacrer beaucoup d'effort pour qu'un modèle statistique puisse utiliser avantageusement cette information.

CHAPITRE 4

CONCLUSION

4.1 Travaux futurs

Comme évoqué en section 3.5, l'implémentation des règles proposées dans Surdeanu *et al.* (2003) pour déterminer le mot le plus représentatif du contenu d'un argument serait une direction naturelle pour poursuivre mon travail.

Parmi tant d'autres améliorations possibles, dans mes expériences, les modèles des cadres sont entrainés séparément, alors que certains cadres sémantiques se ressemblent, et certains rôles sont communs ou similaires parmi différents cadres (voir la discussion en section 2.5). Comme évoqué en section 3.6, l'étiquetage joint des arguments d'un cadre pourrait également être davantage exploré.

Du point de vue de l'apprentissage machine, les modèles pourraient être mieux entrainés. Un ensemble de développement permettrait de mieux adapter les paramètres à chaque méthode (fixer la valeur de k séparément pour chaque cadre, et inclure ou non certaines caractéristiques de base). Un réseau de neurones plus complexe que le modèle à maximum d'entropie utilisé permettrait peut-être de mieux combiner les caractéristiques, plutôt que de simplement leur assigner des poids.

Enfin, il serait intéressant d'utiliser d'autres représentations distribuées de mots (entrainées par d'autres systèmes que SENNA), pour comparer les résultats.

4.2 Conclusion

FrameNet définit un ensemble de cadres sémantiques appelés par des prédicats, ainsi que les rôles pouvant être remplis par les arguments du dit prédicat. J'ai employé des représentations distribuées de mots, entrainées par SENNA, pour

améliorer la tâche de classification des arguments en rôles, en supposant connus le prédicat, le cadre sémantique, et la position des arguments.

Les représentations de mots situent les mots du lexique, *via* leurs coordonnées, dans un espace, l'espace des représentations. La représentation des arguments à classifier par leur mot le plus représentatif – dans mes expériences, leur tête syntaxique – permet de les situer eux-mêmes dans l'espace des représentations. Dès lors, on peut utiliser l'algorithme des plus proches voisins, ou bien partitionner l'espace suivant le rôle *en moyenne* le plus proche, pour classifier les arguments, et faire des prédictions raisonnables, un peu inférieures à l'état de l'art.

En utilisant ces prédictions dans le cadre d'un modèle à maximum d'entropie, on améliore les performances d'un modèle de référence utilisant uniquement des caractéristiques de l'argument dans la phrase, et on dépasse du même coup le système à l'état de l'art SEMAFOR sur la tâche évaluée.

En particulier, les représentations de mots permettent de mieux généraliser les données d'entrainement, *via* le lexique. Le modèle de référence performe naturellement moins bien sur les cadres avec moins d'exemples d'entrainement, et l'information complémentaire apportée permet de réhausser les performances de ces cadres (ce que la macro-précision met en évidence en accordant la même valeur à tous les cadres, quel que soit leur nombre d'exemples).

BIBLIOGRAPHIE

Baker, Collin, Ellsworth, Michael, & Erk, Katrin. 2007. SemEval'07 task 19 : frame semantic structure extraction. *Pages 99–104 of : Proceedings of the 4th International Workshop on Semantic Evaluations*. SemEval '07. Stroudsburg, PA, USA : Association for Computational Linguistics.

Carreras, Xavier, & Màrques, Lluís. 2004. Introduction to the CoNLL-2004 Shared Task : Semantic Role Labeling. *Pages 89–97 of : Proceedings of CoNLL-2004*. Boston, MA, USA.

Carreras, Xavier, & Màrquez, Lluís. 2005. Introduction to the CoNLL-2005 shared task : semantic role labeling. *Pages 152–164 of : Proceedings of the Ninth Conference on Computational Natural Language Learning*. CONLL '05. Stroudsburg, PA, USA : Association for Computational Linguistics.

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

Collobert, Ronan, & Weston, Jason. 2008. A unified architecture for natural language processing : deep neural networks with multitask learning. *Pages 160–167 of :* Cohen, William W., McCallum, Andrew, & Roweis, Sam T. (eds), *ICML*. ACM International Conference Proceeding Series, vol. 307. ACM.

Coppola, Bonaventura, & Moschitti, Alessandro. 2010. A General Purpose FrameNet-based Shallow Semantic Parser. *In : Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. Valletta, Malta : European Language Resources Association (ELRA).

Cover, T., & Hart, P. 1967. Nearest neighbor pattern classification. **13**, 21– 27.

Das, Dipanjan, & Smith, Noah A. 2011. Semi-supervised frame-semantic parsing for unknown predicates. *Pages 1435–1444 of : Proceedings of the 49th Annual*

Meeting of the Association for Computational Linguistics : Human Language Technologies - Volume 1. HLT '11. Stroudsburg, PA, USA : Association for Computational Linguistics.

Das, Dipanjan, Schneider, Nathan, Chen, Desai, & Smith, Noah A. 2010. Probabilistic frame-semantic parsing. *Pages 948–956 of : Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT '10. Stroudsburg, PA, USA : Association for Computational Linguistics.

Das, Dipanjan, Martins, André F. T., & Smith, Noah A. 2012. An Exact Dual Decomposition Algorithm for Shallow Semantic Parsing with Constraints. *Pages 209–217 of : Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1 : Proceedings of the Main Conference and the Shared Task, and Volume 2 : Proceedings of the Sixth International Workshop on Semantic Evaluation*. SemEval '12. Stroudsburg, PA, USA : Association for Computational Linguistics.

de Marneffe, Marie-Catherine, MacCartney, Bill, & Manning, Christopher D. 2006. Generating typed dependency parses from phrase structure parses. *Pages 449–454 of : Proceedings of the International conference on Language Resources and Evaluation (LREC'10)*.

Erk, Katrin, & Pado, Sebastian. 2006. Shalmaneser - a flexible toolbox for semantic role assignment. *Pages 527–532 of : Proceedings of LREC 2006*.

Fillmore, Charles J. 1976. Frame Semantics and the Nature of Language. *Annals of the New York Academy of Sciences*, **280**(1), 20–32.

Fleischman, Michael, & Hovy, Eduard. 2003. A maximum entropy approach to FrameNet tagging. *Pages 22–24 of : Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on*

Human Language Technology : companion volume of the Proceedings of HLT-NAACL 2003–short papers - Volume 2. NAACL-Short '03. Stroudsburg, PA, USA : Association for Computational Linguistics.

Fleischman, Michael, Kwon, Namhee, & Hovy, Eduard. 2003. Maximum Entropy Models for FrameNet Classification. *Pages 49–56 of : Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. EMNLP '03. Stroudsburg, PA, USA : Association for Computational Linguistics.

Gildea, Daniel, & Jurafsky, Daniel. 2000. Automatic Labeling of Semantic Roles. *In : Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*. ACL.

Johansson, Richard, & Nugues, Pierre. 2007. LTH : semantic structure extraction using nonprojective dependency trees. *Pages 227–230 of : Proceedings of the 4th International Workshop on Semantic Evaluations*. SemEval '07. Stroudsburg, PA, USA : Association for Computational Linguistics.

Kingsbury, Paul, & Palmer, Martha. 2002. From Treebank to Propbank. *In : Language Resources and Evaluation*.

Kingsbury, Paul, & Palmer, Martha. 2003. PropBank : The next level of TreeBank. *In : Proceedings of Treebanks and Lexical Theories*.

Koomen, P., Punyakanok, V., Roth, D., & Yih, W. 2005. Generalized Inference with Multiple Semantic Role Labeling Systems Shared Task Paper. *Pages 181–184 of :* Dagan, Ido, & Gildea, Dan (eds), *CoNLL*.

Kotsiantis, S. B. 2007. *Supervised Machine Learning : A Review of Classification Techniques. Informatica 31 :249–268*.

Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. *Pages 768–774 of : Proceedings of the 36th Annual Meeting of the Association for*

Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2. ACL '98. Stroudsburg, PA, USA : Association for Computational Linguistics.

Màrquez, Lluís, Carreras, Xavier, Litkowski, Kenneth C., & Stevenson, Suzanne. 2008. Semantic Role Labeling : An Introduction to the Special Issue. *Computational Linguistics*, **34**(2), 145–159.

Palmer, Martha, Gildea, Daniel, & Kingsbury, Paul. 2005. The Proposition Bank : An Annotated Corpus of Semantic Roles. *Computational Linguistics*, **31**(1), 71–106.

Punyakanok, V., Roth, D., & Yih, W. 2008. The Importance of Syntactic Parsing and Inference in Semantic Role Labeling. *Computational Linguistics*, **34**(2).

Ratnaparkhi, A. 1997. *A simple introduction to maximum entropy models for natural language processing*. Tech. rept. Institute for Research in Cognitive Science, University of Pennsylvania.

Surdeanu, Mihai, Harabagiu, Sanda, Williams, John, & Aarseth, Paul. 2003. Using predicate-argument structures for information extraction. *Pages 8–15 of : Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*. ACL '03. Stroudsburg, PA, USA : Association for Computational Linguistics.

Tan, Songbo. 2005. Neighbor-weighted K-nearest Neighbor for Unbalanced Text Corpus. *Expert Syst. Appl.*, **28**(4), 667–671.

Turian, Joseph, Ratinov, Lev, & Bengio, Yoshua. 2010. Word representations : a simple and general method for semi-supervised learning. *Pages 384–394 of : Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Stroudsburg, PA, USA : Association for Computational Linguistics.

Wagner, Wiebke. 2010. Steven Bird, Ewan Klein and Edward Loper : Natural Language Processing with Python, Analyzing Text with the Natural Language Toolkit. *Lang. Resour. Eval.*, **44**(4), 421–424.

Xue, Nianwen, & Palmer, Martha. 2004. Calibrating Features for Semantic Role Labeling. *Pages 88–94 of : EMNLP.* ACL.