Modèles exponentiels et contraintes sur les espaces de recherche en traduction automatique et pour le transfert cross-lingue

Nicolas Pécheux Pécheux

NNT : 2016SACLS242

THÈSE DE DOCTORAT
DE
L'UNIVERSITÉ PARIS-SACLAY
PRÉPARÉE À
L'UNIVERSITÉ PARIS-SUD
ET AU
LABORATOIRE D'INFORMATIQUE POUR LA MÉCANIQUE ET LES SCIENCES DE L'INGÉNIEUR

ÉCOLE DOCTORALE N°580
Sciences et technologies de l'information et de la communication

Spécialité de doctorat : Informatique

Par

M. Nicolas Pécheux

Modèles exponentiels et contraintes sur les espaces de recherche en traduction automatique et pour le transfert cross-lingue

Thèse présentée et soutenue à Orsay, le 27 septembre 2016 :

Composition du Jury :

Mme Anne Vilnat, Professeure, Université Paris-Saclay, Présidente du jury
Mme Isabelle Tellier, Professeure, Université Paris 3, Rapporteuse
M. Fabrice Lefèvre, Professeur, Université d'Avignon et des Pays de Vaucluse, Rapporteur
M. Massih-Reza Amini, Professeur, Examinateur
M. François Yvon, Professeur, Université Paris-Saclay, Directeur de thèse
M. Alexandre Allauzen, Maître de Conférence, Université Paris-Saclay, Co-encadrant de thèse

À Coline,

Remerciements et gratitude pour la réalisation de la présente thèse

Merci pour l'accueil
à l'équipe Traitement du Langage Parlé
du LIMSI-CNRS
et à l'Université Paris-Sud
https://www.universite-paris-saclay.fr

Résumé

Mon travail n'aurait pu aboutir sans le concours de nombreuses personnes. Cet article met en lumière ceux qui m'ont apporté leur soutien ou leur collaboration, partageant avec moi cette tranche de vie.

1 Introduction

J'aurais difficilement imaginé de meilleurs encadrants que mes deux directeurs, François YVON et Alexandre ALLAUZEN. Depuis mon arrivée au LIMSI, Alexandre n'a cessé de me consacrer de longues séances de travail, à décrypter des codes, à réfléchir à de nouvelles idées ou à finaliser des articles, avec une formidable gestion de l'urgence. Je garde d'excellents souvenirs tout aussi marquants de mes innombrables échanges scientifiques avec François. Je me rappelle que chaque document soumis me revenait dès le lendemain entièrement annoté de questions, de commentaires et de nouvelles pistes, donnant lieu à d'éclairantes discussions. Je leur suis extrêmement reconnaissant de m'avoir fait découvrir — et surtout apprécier — le monde de la recherche. Qu'ils trouvent ici l'expression de ma profonde gratitude pour leur patience et leur bienveillance, et pour n'avoir jamais cessé de m'accorder leur confiance.

Je souhaite également remercier Isabelle TELLIER et Fabrice LEFÈVRE pour leur rapport et pour la relecture et l'annotation minutieuse de mon manuscrit ; Massih-Reza AMINI pour son intérêt pour mon travail et sa participation à mon jury ; et Anne VILNAT pour avoir accepté de présider mon jury de soutenance et pour l'établissement du rapport final.

2 Méthode : laboratoire et collègues

La figure 1 illustre le laboratoire dans lequel j'ai eu la chance d'effectuer mes travaux de recherche.

FIGURE 1 – Merci au LIMSI de m'avoir offert d'excellentes conditions de travail, dans une ambiance chaleureuse qui va beaucoup me manquer.

2.1 Collaborateurs principaux

Deux collègues ont joué un rôle majeur dans ma formation à la recherche et dans mon travail de thèse :

- **Thomas LAVERGNE** Une partie de mes recherches ont été menées en collaboration avec Thomas, même si tout ne figure pas dans ce manuscrit. Il m'a très vite initié aux questionnements propres à l'enseignement et à la recherche, et je le remercie également pour son implémentation efficace des CRF, largement utilisée dans mes travaux.
- **Guillaume WISNIEWSKI** On m'avait prévenu qu'il n'y avait pas de meilleur cobureau que lui. Il m'est en effet impossible de retranscrire tout ce que m'ont apporté ses précieux conseils et ses réflexions, tout au long de ma thèse. Le troisième chapitre est le fruit de l'une de nos collaborations.

2.2 Collègues

Je remercie Hervé pour son esprit collaboratif et pour m'avoir invité à rejoindre son projet. Merci à Hélène pour son amitié, à Aurélien pour nos longues et passionnantes discussions, et à toute l'équipe du LIMSI [1, 15] pour avoir rendu mon quotidien aussi agréable et stimulant.

2.3 Doctorants [1]

$$\sum_{\mathrm{Li}=\mathrm{Yong}^{2}}^{(\mathrm{Khanh})!} \frac{\mathrm{Elena}_{\mathrm{Nadi}}(\mathrm{Penny}|\overline{\mathrm{Thiago}})}{\pi\mathrm{Lauriane}^{\mathrm{Souhir}-\mathrm{Hai\ Son}}} = \left| \begin{array}{cc} \frac{\mathrm{Clement}}{\mathrm{Kevin}} & \Psi(\mathrm{Rachel}) \\ \mathrm{Julia}^{\alpha} & \frac{\mathrm{Matthieu}}{\mathrm{Benjamin}} \end{array} \right|$$

1. Un grand nombre d'entre eux sont aujourd'hui docteurs. Les autres le seront prochainement.

Des remerciements amicaux à la joyeuse et dynamique équipe des thésards pour nos nombreux échanges, discussions et projets.

3 Expériences familiales

De chaleureux remerciements à ma mère, qui m'a relu plus d'une fois, à mon père qui m'a initié à l'informatique mais qui n'a rien compris à mon travail, à ma sœur, qui n'a jamais su expliquer ce que je faisais exactement, ainsi qu'à toute ma famille [11]. Mes pensées vont également à ma grand-mère, Hélène, qui n'a pas pu voir l'achèvement de ces travaux.

4 Expériences amicales

Je remercie mes amis pour leur soutien et leur affection [5, 7, 8, 10, 13] et pour nos activités partagées [2, 3, 9, 12, 14], en particulier Pierre que je n'ai pas pu voir assez souvent, Annabelle qui a terminé sa thèse bien avant moi et Marine V. qui devait enregistrer mon dépôt de thèse.

J'exprime toute ma gratitude et mon amitié à Anne-Cécile, Julie, Lino et Marine O. pour s'être aussi bien occupé de moi pendant toutes ces années [6].

5 Résultats

5.1 NLP researcher

Many thanks to Aylien [2] for the exciting job opportunity in an amazing startup and for the provided flexibility at the end of my Ph.D. [4].

5.2 Perspectives

Merci aux collègues du Lycée Carnot pour leur accueil et en particulier à Jérémy pour nos $2^{8.4}$ courriels échangés et pour ses précieux conseils.

6 Conclusion

En guise de conclusion, je propose l'exécution [3] de l'algorithme 1. Enfin, je remercie tout particulièrement Coline pour son soutien indéfectible, son énergie et son aide si précieuse tout au long du déroulement de ce projet.

Acknowledgments

This thesis was supported by a government ASN fellowship. I also thank Université Paris-Sud for a two-months funding extention.

2. `http://aylien.com`

3. La complexité en temps de cet algorithme peut être assez élevée.

Algorithme 1 : Remerciements généraux.

Entrée : L'ensemble P des personnes qui m'ont permis — directement ou indirectement — de mener à bien ce travail.

```
pour p dans P faire
    si p n'est pas explicitement citée alors
        remercier(p);
    fin
fin
```

Références

[1] Abderahman, Alexandre, AMIC, Anh-Phuong, Anne, Anindya, Antoine, Artem, Aurélien, Bénédicte, Benjamin, Bill, Camille, Claude, Clément, David, Elena, Éric, Fan, Franck, François, Gilles, Gregory, Guillaume, Hai Son, Hélène, Hervé, Ilya, Jean-Luc, Johann, Julia, Kevin, Khanh, Laurence D., Laurence R., Lauriane, Li, Linlin, Lori, Lucile, Marco, Marianna, Mariette, Matthieu, Nada, Nadège, Nadi, Natalia, Nicole, Ophélie, Pascal, Penny, Philippe, Pierre, Rachel, Rita, Sophie, Souhir, Stéphanie, Théodore, Thiago, Thomas, Vincent, Yong. In *LIMSI*, Orsay, 2012–2016.

[2] Adeline, Alban, Aline, Alise, Caroline, Cassandre, Cecilia, Émilien, Fabien, Fanny, Ilsée, Ilya, Janniek, Joana, Karine, Laeticia, Laurent, Louise, Lúcia, Magdalena, Margot, Marion, Maxime, Naomi, Nastasia, Nicolas, Patrick, Filipe, Rian, Sarah, Zoé. In *Folk*, Everywhere, 2008–2016.

[3] Adeline, son mari, ses parents. In *Ski*, Flumet, 2012–2013.

[4] Afshin, Amir, Chris, Hamed, John, Kevin, Mike, Noel, Parsa, Peiman, Robson, Sebastian. Team. In *Aylien*, Dublin, 2016.

[5] Agathe, Annabelle, Anne, Aude, Cecilia, Elena, Fede, Korrigan, Laura, Nounou, Maël, Marianne, Marine, Marion, Pierre, Romain, Xavier. In *Friends*.

[6] Alex, Anne-Cécile, Antoine, Clara, Clément, Diane, Errel, Jane, Julie, Lino, Marine, Sofia. Colocs. In *Blumhaus*, Fontenay-aux-Roses, 2011–2015.

[7] Anne-Thérèse, Cyril. Folk. In *Le petit chat Production*, Paris, 2015.

[8] Antoine, Coline, Marine, Sylvain, Vincent. Synchronistes. In *Zikmu*, 2012–2016.

[9] Blandine, Carolina, Claudia, Danielle, Federico, Gwenaëlle, Jean-Paul, Julien, Marie-Olivia, Romain. Tango. In *Tango Ostinato*, Paris, 2010–2014.

[10] Camille, Clarisse, Clémentine, Édouart, Guibé, Marion, Maÿlis, Romain, Sarah, Yves. In *Furets*, 2005–2016.

[11] Camille, Daniel, France, Hélène, Jeff, Lucille, Marie, Marianne, Mathilde, Michel, Olivier, Patricia, Pauline, Raymonde, Soizïck. In *Famille*.

[12] Candice, Carlos, Fabienne, Jean-Say, Marion, Myriam, Soline. Forró. In *Le P'tit Bal Perdu*, Paris, 2013–2016.

[13] Chiara, Claire, Léo, Nicolas, Timothée, Théophile, Sasha. In *Cogmaster*, Paris, 2010–2016.

[14] Didier. In *Yoga*, Paris, 2011–2013.

[15] Équipe de foot. In *LIMSI*, Orsay, 2012–2014.

Résumé

La plupart des méthodes de traitement automatique des langues (TAL) peuvent être formalisées comme des problèmes de prédiction, dans lesquels on cherche à choisir automatiquement l'hypothèse la plus plausible parmi un très grand nombre de candidats. Malgré de nombreux travaux qui ont permis de mieux prendre en compte la structure de l'ensemble des hypothèses, la taille de l'espace de recherche est généralement trop grande pour permettre son exploration exhaustive. Dans ce travail, nous nous intéressons à l'importance du design de l'espace de recherche et étudions l'utilisation de contraintes pour en réduire la taille et la complexité. Nous nous appuyons sur l'étude de trois problèmes linguistiques — l'analyse morpho-syntaxique, le transfert cross-lingue et le problème du réordonnancement en traduction — pour mettre en lumière les risques, les avantages et les enjeux du choix de l'espace de recherche dans les problèmes de TAL.

Par exemple, lorsque l'on dispose d'informations *a priori* sur les sorties possibles d'un problème d'apprentissage structuré, il semble naturel de les inclure dans le processus de modélisation pour réduire l'espace de recherche et ainsi permettre une accélération des traitements lors de la phase d'apprentissage. Une étude de cas sur les modèles exponentiels pour l'analyse morpho-syntaxique montre paradoxalement que cela peut conduire à d'importantes dégradations des résultats, et cela même quand les contraintes associées sont pertinentes. Parallèlement, nous considérons l'utilisation de ce type de contraintes pour généraliser le problème de l'apprentissage supervisé au cas où l'on ne dispose que d'informations partielles et incomplètes lors de l'apprentissage, qui apparaît par exemple lors du transfert cross-lingue d'annotations. Nous étudions deux méthodes d'apprentissage faiblement supervisé, que nous formalisons dans le cadre de l'apprentissage ambigu, appliquées à l'analyse morpho-syntaxiques de langues peu dotées en ressources linguistiques.

Enfin, nous nous intéressons au design de l'espace de recherche en traduction automatique. Les divergences dans l'ordre des mots lors du processus de traduction posent un problème combinatoire difficile. En effet, il n'est pas possible de considérer l'ensemble factoriel de tous les réordonnancements possibles, et des contraintes sur les permutations s'avèrent nécessaires. Nous comparons différents jeux de contraintes et explorons l'importance de l'espace de réordonnancement dans les performances globales d'un système de traduction. Si un meilleur design permet d'obtenir de meilleurs résultats, nous montrons cependant que la marge d'amélioration se situe principalement dans l'évaluation des réordonnancements plutôt que dans la qualité de l'espace de recherche.

Mots clefs : Traduction automatique ; Contraintes de réordonnancement ; Étiquetage morpho-syntaxique ; Transfert cross-lingue ; Apprentissage faiblement supervisé ; Champs markoviens aléatoires

Abstract

Most natural language processing tasks are modeled as prediction problems where one aims at finding the best scoring hypothesis from a very large pool of possible outputs. Even if algorithms are designed to leverage some kind of structure, the output space is often too large to be searched exaustively. This work aims at understanding the importance of the search space and the possible use of constraints to reduce it in size and complexity. We report in this thesis three case studies which highlight the risk and benefits of manipulating the seach space in learning and inference.

When information about the possible outputs of a sequence labeling task is available, it may seem appropriate to include this knowledge into the system, so as to facilitate and speed-up learning and inference. A case study on type constraints for CRFs however shows that using such constraints at training time is likely to drastically reduce performance, even when these constraints are both correct and useful at decoding.

On the other side, we also consider possible relaxations of the supervision space, as in the case of learning with latent variables, or when only partial supervision is available, which we cast as ambiguous learning. Such weakly supervised methods, together with cross-lingual transfer and dictionary crawling techniques, allow us to develop natural language processing tools for under-resourced languages.

Word order differences between languages pose several combinatorial challenges to machine translation and the constraints on word reorderings have a great impact on the set of potential translations that is explored during search. We study reordering constraints that allow to restrict the factorial space of permutations and explore the impact of the reordering search space design on machine translation performance. However, we show that even though it might be desirable to design better reordering spaces, model and search errors seem yet to be the most important issues.

Key words : Statistical Machine Translation ; Reordering Contraints ; Cross-Lingual Transfer ; Weakly Supervised Learning ; Conditional Random Fields

Table des matières

Chapitre 1

Introduction

Sommaire

L'intérêt de la traduction automatique (TA) et du traitement automatique des langues (TAL) n'est plus à démontrer tant leurs outils sont intégrés dans notre quotidien. Aujourd'hui, sur Internet, il est possible d'exprimer des requêtes directement dans notre langue naturelle — par opposition à l'utilisation d'une syntaxe spécifique suivant un format prédéfini — pour obtenir quasiment instantanément une réponse profilée suivant nos interactions précédentes, des suggestions similaires, et même des informations provenant d'une source rédigée dans une langue autre que la nôtre. Il n'est pas rare de lire un article de presse, la confirmation d'une réservation ou la description d'un produit dans notre langue sans sourciller et, devant quelques tournures un peu étranges, de se rendre compte que le texte vient en fait d'être instantanément traduit automatiquement à partir d'une autre langue. Pourtant, les langues naturelles consituent un phénomène complexe difficile à décrire dans le plus fin détail, et *a fortiori* difficile à modéliser. Décrire ou modéliser une langue ne peut se faire simplement à l'aide de règles, tant les variations linguistiques sont importantes et les exceptions nombreuses. Le projet de doter une machine de capacités linguistiques en exploitant ces modèles reste à ce jour inabouti, en dépit de nombreuses tentatives. Parmi ces applications, la traduction, qui consiste à reformuler dans une langue ce qui est exprimé dans une

autre, nécessite une compréhension profonde du sens et ne saurait se réduire à un simple traitement de surface (Yngve, 1957). Pour certaines applications — la traduction de textes de loi ou de contrats d'entreprises par exemple — la qualité de la traduction est souvent primordiale (Bellos, 2012) et cette dernière ne peut donc pas être confiée intégralement à des machines. Dans d'autres cas, pour comprendre globalement le contenu d'un article de presse en langue étrangère par exemple, la traduction automatique peut fournir des réponses acceptables, et le nombre toujours plus grand de telles requêtes rend aujourd'hui nécessaire un traitement automatique. Des progrès considérables ont été rendus possibles par la conjonction de deux événements importants. D'une part, l'avènement d'Internet a massivement contribué à constituer, répertorier et partager de grands volumes de données textuelles, et ce dans plusieurs langues. Il a ainsi été possible, par exemple, de constituer d'importants corpus parallèles, c'est-à-dire de phrases en relation de traduction mutuelle, ce qui fournit un très grand nombre d'exemples de traductions humaines. D'autre part, ces dernières décennies ont donné lieu à d'importants développements de méthodes statistiques pour réaliser des applications concrètes de TAL, qui ont peu à peu supplanté les méthodes à base de règles manuelles, peu robustes lorsqu'il s'agit de passer à l'échelle. Ces méthodes utilisent les exemples — et sont d'autant plus efficaces que ces exemples sont nombreux — pour collecter des statistiques qui permettent ensuite de choisir de manière probabiliste les meilleures décisions, et ce pour des applications très variées.

S'il n'est généralement pas possible d'expliciter les règles qui régissent les phénomènes linguistiques, on peut en revanche, dans certains cas, les modéliser approximativement, en utilisant la régularité et les motifs implicites des structures linguistiques sous-jacentes. C'est en tout cas l'hypothèse émise par le cadre de l'apprentissage automatique supervisé qui, à partir d'exemples préalablement annotés, permet d'apprendre automatiquement les mécanismes qui conduisent d'un exemple à son annotation. La figure 1.1 propose un exemple de tâche d'analyse linguistique. Cette tâche d'analyse morpho-syntaxique, que nous étudions en détail dans la première partie de ce document, consiste à savoir prédire, pour un mot donné d'une phrase, son étiquette morpho-syntaxique. Si l'on dispose d'un corpus annoté, c'est-à-dire des étiquettes de référence, il est possible d'apprendre de manière supervisée un modèle d'étiquetage qui pourra inférer des séquences d'étiquettes associées à de nouvelles phrases. Formellement, dans un problème d'apprentissage générique, on dispose d'un ensemble $\mathcal{X}$ des *entrées* possibles, ici des phrases dans une langue, ainsi que celui $\mathcal{Y}$ des *sorties* possibles, ici l'ensemble des analyses morpho-syntaxiques de ces phrases. On parle d'apprentissage structuré, lorsque l'ensemble $\mathcal{Y}$ est complexe et lorsque l'on décompose ce dernier lors du processus de modélisation. Dans le cas de l'analyse morpho-syntaxique par exemple, l'espace de sortie est structuré sous la forme d'une séquence d'étiquettes.

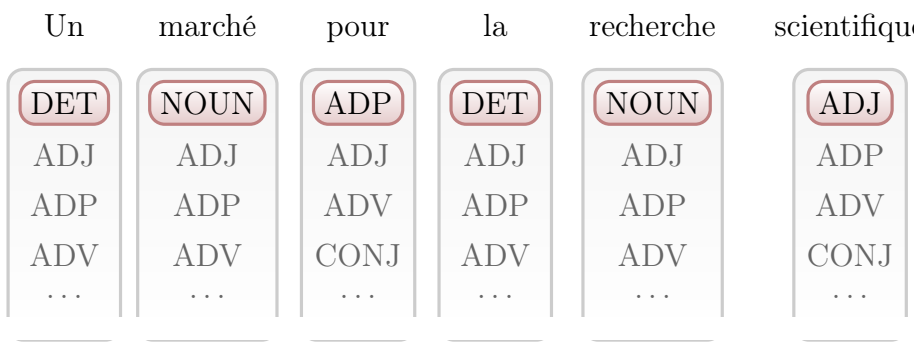


FIGURE 1.1 – L'analyse morpho-syntaxique vue comme un problème d'étiquetage structuré. À chaque mot est associée une étiquette de référence (en rouge) parmi les étiquettes possibles (en gris). C'est cette séquence d'étiquettes que l'on cherche à prédire.

On peut également considérer le problème de la traduction comme un problème d'apprentissage structuré. À partir d'une phrase source $\mathbf{x} \in \mathcal{X}$, où $\mathcal{X}$ est alors l'ensemble des phrases de la langue source, on cherche à associer « sa » traduction $\mathbf{y} \in \mathcal{Y}$, où $\mathcal{Y}$ est l'ensemble des phrases en langue cible.

1.1 Modèles exponentiels

Nous utilisons et étudions principalement les modèles exponentiels qui définissent une distribution de probabilité sur les sorties $\mathbf{y} \in \mathcal{Y}$ possibles, par exemple des phrases en langue cible, connaissant la phrase en langue source $\mathbf{x} \in \mathcal{X}$ comme :

$$p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x})} \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})\right) \tag{1.1}$$

où $\phi(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^d$ est un vecteur de d caractéristiques, $\boldsymbol{\theta} \in \mathbb{R}^d$ un vecteur de paramètres et

$$\mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})\right) \tag{1.2}$$

est le terme de normalisation. On peut ensuite, pour une entrée $\mathbf{x} \in \mathcal{X}$ donnée, prédire la sortie associée en prenant par exemple l'hypothèse la plus probable selon le modèle. L'enjeu principal est alors de choisir des caractéristiques qui permettent de définir un bon modèle et pour lesquelles on pourra apprendre, automatiquement, les paramètres de poids associés.

Le vecteur de caractéristiques $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$ peut, en théorie, considérer n'importe quel aspect de la phrase source $\mathbf{x}$ et de la phrase cible $\mathbf{y}$. Pour des raisons com-

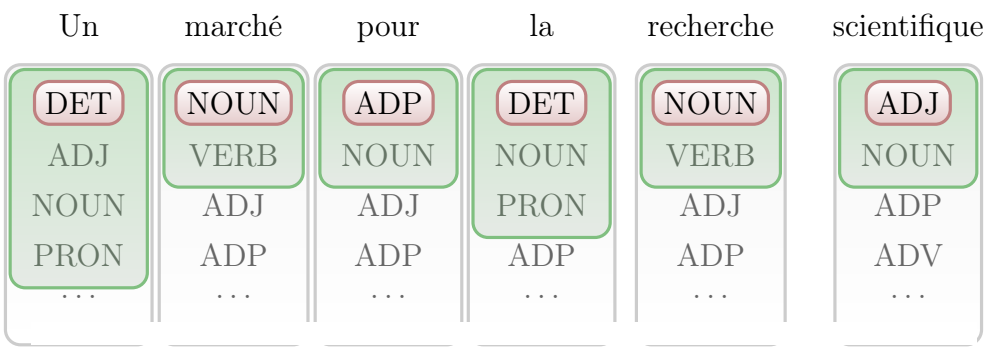


FIGURE 1.2 – Contraintes de dictionnaire (en vert) lors de l'analyse morpho-syntaxique, qui permettent de réduire l'espace de recherche et, on l'espère, de simplifier la tâche du modèle par l'introduction de connaissances linguistiques.

putationnelles, il est nécessaire de limiter les caractéristiques sur les sorties $\mathbf{y}$ en utilisant la structure de l'espace $\mathcal{Y}$ et/ou de recourir à une inférence approchée.

1.2 Contraintes sur l'espace de recherche

Dans de nombreux cas, même avec des algorithmes adaptés à la structure de l'espace de sortie, utilisant par exemple la programmation dynamique, le calcul de la somme de l'équation (1.2) peut se révéler prohibitif. En traduction automatique par exemple, la somme (1.2) implique de pouvoir sommer sur toutes les phrases de la langue cible. Toujours pour des raisons computationnelles, il est donc souvent nécessaire de limiter le nombre d'hypothèses que l'on peut explorer, par exemple le nombre de traductions que peut prendre un segment de phrase donné. Au lieu de considérer l'espace $\mathcal{Y}$ en entier, on va donc s'intéresser, pour un $\mathbf{x} \in \mathcal{X}$, à un espace restreint $\mathcal{Y}(\mathbf{x})$. Des contraintes sur l'espace de recherche peuvent intervenir dans de nombreuses tâches où le nombre d'étiquettes serait trop grand, comme par exemple l'analyse morpho-syntaxique de langues morphologiquement riches, par exemple l'allemand, que nous étudions dans ce travail. Plus généralement, nous nous interrogeons sur l'importance de l'espace de recherche et la possibilité d'utiliser des contraintes pour en réduire la taille et la complexité. Dans certains cas, de telles contraintes peuvent s'avérer nécessaires, par exemple pour définir l'espace de recherche en traduction automatique en raison des nombreux réordonnancements possibles, ce que nous étudions dans la deuxième partie de ce manuscrit. Dans d'autre cas, les contraintes peuvent également traduire une forme de connaissance linguistique préalable. Lorsque l'on dispose de connaissances *a priori* sur les sorties possibles d'un problème d'étiquetage, il semble en effet pertinent d'inclure cette information lors de l'apprentissage pour simplifier la tâche de modélisation et ac-

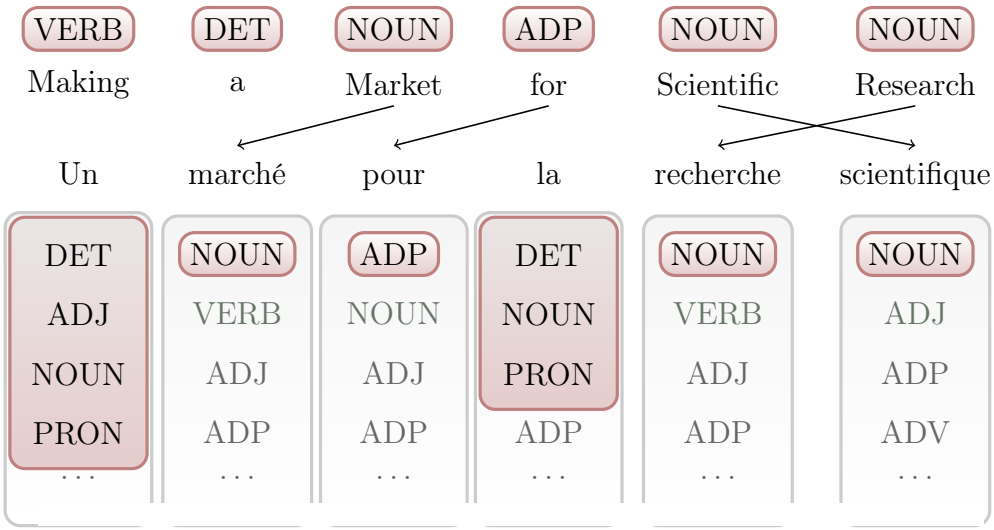


FIGURE 1.3 – Transfert cross-lingue vu comme un problème d'apprentissage ambigu. Les étiquettes morpho-syntaxiques de la phrase source sont transférées vers la phrase cible. En l'absence de liens d'alignements on peut utiliser les contraintes de dictionnaire. Voir la section 3.3.5 pour plus de détails. Des erreurs d'alignements (ici le lien entre « a » et « Un » est manquant) ou d'étiquetage de la phrase source (ici « Scientific » devrait être un adjectif) sont possibles.

célérer les traitements. Nous étudions l'effet de telles contraintes dans le cadre de l'analyse morpho-syntaxique lorsque nous disposons d'un dictionnaire associant à chaque mot l'ensemble des étiquettes morpho-syntaxiques que peut prendre ce mot. La figure 1.2 illustre cela par un exemple. Paradoxalement, nous observons que de telles contraintes peuvent dégrader sévèrement les performances lorsqu'elles sont intégrées lors de l'apprentissage, et ce même lorsqu'elles sont correctes et bénéfiques au décodage. Dans le quatrième chapitre de ce document, nous étudions ce paradoxe et montrons que cet effet indésirable est dû à un manque de contraste dans les sorties possibles et entraîne une forme de surapprentissage, particulièrement important lorsque les contraintes permettent de désambiguïser complètement un mot. Nous proposons quelques solutions simples pour limiter ce problème.

1.3 Apprentissage ambigu et transfert cross-lingue

Nous utilisons également ce type de contraintes, de manière un peu différente, pour réduire l'espace des étiquettes possibles dans le cadre de l'apprentissage partiellement supervisé, avec comme exemple d'application l'analyse morpho-

syntaxique de langues peu dotées en ressources. En effet, si l'apprentissage supervisé a permis de grandes avancées pendant ces dernières décennies, il est tributaire de la disponibilité de grandes quantités de données annotées manuellement. Le corpus arboré Penn Treebank [1] (Marcus *et al.*, 1994), contenant à ce jour plusieurs millions de mots annotés, est peut-être l'exemple le plus connu de ressource et a permis avec succès la réalisation de nombreuses études. Mais de telles ressources sont rares, en particulier pour de nombreuses langues moins étudiées. Afin d'apprendre un analyseur syntaxique pour des langues peu dotées en ressources linguistiques, une approche possible est de transférer automatiquement, mais seulement partiellement, les annotations d'une autre langue riche en ressources en exploitant des corpus parallèles alignés automatiquement au niveau des mots (Yarowsky *et al.*, 2001). Ceci est illustré par la figure 1.3. En combinant cette technique avec des dictionnaires automatiquement extraits de Wiktionnaire (Li *et al.*, 2012a) ou directement des corpus parallèles, on peut restreindre de manière importante les étiquettes possibles pour chaque mot (Täckström *et al.*, 2013a). Mais cette information de supervision reste partielle, nous dirons *ambiguë*, et constitue un cadre moins favorable que celui de l'apprentissage supervisé, tel que le montre la figure 1.2. Dans le troisième chapitre, nous étudions et comparons deux méthodes différentes pour apprendre à partir d'informations ambiguës, évaluons l'importance des différentes sources de contraintes et concluons sur les limites de l'évaluation d'une telle tâche.

1.4 Traduction automatique

Le processus de traduction permettant de passer d'une phrase source à sa traduction en langue cible est complexe et difficile à modéliser directement. Contrairement au problème, également structuré, de l'analyse morpho-syntaxique, il n'apparaît pas de structure évidente suivant laquelle il serait possible de décomposer le processus. De nombreuses hypothèses ont été proposées : la traduction mot à mot, les modèles à base de segments, les modèles hiérarchiques ou les modèles syntaxiques. Dans l'approche à base de segments, on peut considérer à quelques variantes près que le processus de traduction se décompose en trois étapes : (a) une étape de segmentation, qui consiste à découper la phrase source en unités formées de segments de mots ; (b) la permutation de ces segments pour rendre compte des divergences éventuelles dans l'ordre des unités, étape que l'on appelle le *réordonnancement* ; et (c) la traduction de ces segments de mots sources préalablement réordonnés en segments de mots cibles. Cette structure sous-jacente permet à la fois de définir l'*espace de recherche* des traductions d'une phrase qu'il est nécessaire de limiter, et la forme que peuvent prendre les différents modèles.

1. https://www.cis.upenn.edu/~treebank/

En théorie, on considère à partir d'une phrase source $\mathbf{x}$ l'ensemble de toutes les phrases de la langue cible comme des hypothèses envisageables. Cet ensemble est infini, et les approches de traduction automatique définissent un espace de recherche $\mathcal{Y}(\mathbf{x}) \subset \mathcal{Y}$ des traductions possibles pour une phrase $\mathbf{x}$. La forme précise de ces espaces dépend du paradigme choisi, mais on peut considérer qu'un système de traduction comporte deux modules : (a) un mécanisme qui permet de construire l'espace de recherche exploré ; (b) un modèle pour évaluer les différentes hypothèses (Auli *et al.*, 2009). Dans ce travail, nous considérons exclusivement une approche à base de segments, ce qui conduit à une forme particulière de l'espace de recherche, mais nous pensons que la plupart des idées générales développées s'appliqueraient également pour d'autres types de modèles.

1.5 Le problème de l'espace de réordonnancement

Comme tous les systèmes de traduction, le modèle que nous utilisons a besoin d'un mécanisme pour modéliser les changements dans l'ordre des mots. En effet, ce dernier peut sensiblement diverger suivant les langues et n'est souvent pas préservé lors du processus de traduction. Comme il n'est pas possible d'explorer entièrement l'espace factoriel de toutes les permutations des mots, il est nécessaire d'introduire des contraintes pour réduire l'*espace des réordonnancements* possibles. Dans la deuxième partie de ce travail, nous nous intéressons plus particulièrement au problème des réordonnancements et aux diverses méthodes qui, explicitement ou implicitement, permettent de restreindre l'ordre des mots exploré lors de la recherche des hypothèses de traduction. Pour le cas des modèles à base de segments, nous étudions en détail un système à base de règles syntaxiques et montrons ses avantages par rapport à des contraintes purement combinatoires. En effectuant plusieurs analyses oracles, c'est-à-dire qui ont connaissance de l'ordre des mots « correct » pour une phrase donnée, nous mesurons également l'importance de l'espace de réordonnancement et étudions le compromis nécessaire entre son expressivité et sa complexité. Nous concluons que si des gains peuvent être espérés en améliorant son design, ceci doit s'accompagner en premier lieu d'une amélioration dans les modèles pouvant évaluer la pertinence des réordonnancements Cette étude sur l'importance de l'espace de réordonnancement fait l'objet du dernier chapitre de ce manuscrit.

1.6 Contributions

Il nous semble que l'on peut distinguer trois principales contributions dans ce travail de thèse :

- **Apprentissage ambigu :** Une étude poussée de l'utilisation de modèles CRF partiellement supervisés pour l'apprentissage ambigu, une comparaison avec un modèle à base d'historique et l'application des modèles CRF à une tâche de transfert cross-lingue pour les langues peu dotées. Dans ce cadre, nous avons également contribué à une meilleure compréhension de l'importance du choix des contraintes et à l'analyse critique de la tâche et de son évaluation.

- **Contraintes à l'apprentissage :** Une étude détaillée de l'utilisation de contraintes pour réduire l'espace de recherche des CRF, sur les problèmes que cela peut poser lors de l'apprentissage et la proposition de quelques solutions simples.

- **Réordonnancements :** Une étude détaillée du système à base de règles syntaxiques de NCODE[2] qui justifie son utilisation et montre ses limites, ainsi qu'une étude complète sur le compromis et le design de l'espace de réordonnancement qui confirme qu'il est probablement plus important de se concentrer en premier lieu sur la manière dont est évalué l'ordre des mots plutôt que sur l'expressivité de l'espace de recherche.

1.7 Organisation générale

La première partie de ce document traite des modèles exponentiels pour l'apprentissage structuré et s'intéresse à l'utilisation de contraintes pour réduire l'espace de recherche ou pour construire des annotations partielles dans le cadre du transfert cross-lingue pour les langues peu dotées. La deuxième partie s'intéresse à la traduction automatique et étudie divers mécanismes pour générer les différences d'ordre des mots que l'on observe lors du processus de traduction. Nous y évaluons enfin l'importance du design de l'espace de recherche de réordonnancement.

Partie I : Modèles exponentiels pour l'apprentissage structuré

- **Chapitre 2 : Modèles exponentiels pour le traitement automatique des langues**. Ce chapitre détaille les principaux problèmes auxquels nous nous intéressons et introduit les notations que nous allons utiliser tout au long de notre exposé, les principaux modèles et les techniques utilisés. Nous motivons l'intérêt de l'apprentissage discriminant par rapport aux modèles génératifs dans le cas général.

2. http://ncode.limsi.fr

- **Chapitre 3 : Apprentissage ambigu : application au transfert cross-lingue**. Dans ce chapitre nous nous demandons à quel point il est possible d'apprendre lorsque l'on ne dispose pas d'une référence complète mais seulement d'une information ambiguë. Cette question se pose en particulier dans le cadre de l'apprentissage par transfert cross-lingue. Nous étudions en détail l'importance des différentes ressources et alertons sur les problèmes d'évaluation d'un tel cadre. Ce chapitre a fait l'objet d'une collaboration dont une partie a été publiée dans (Wisniewski *et al.*, 2014b) pour les premières expériences, dans (Wisniewski *et al.*, 2014a) pour l'ensemble des dix langues considérées et dans (Pécheux *et al.*, 2016b) pour une étude plus approfondie.

- **Chapitre 4 : Contraintes de types dans les modèles CRF**. Dans ce chapitre, nous examinons en détail ce qui nous semble au premier abord être un paradoxe : pourquoi l'introduction d'événements impossibles dans un modèle de la connaissance peut dégrader sévèrement les performances ? Les observations de ce chapitre ont été également décrites dans (Pécheux *et al.*, 2015).

Partie II : Le choix de l'espace des réordonnancements en traduction automatique

- **Chapitre 5 : Le problème de la traduction automatique**. Dans ce chapitre, nous présentons en détail la traduction automatique et les problématiques associées. C'est également l'occasion de présenter le système état de l'art auquel nous nous comparons et que nous avons continué à développer.

- **Chapitre 6 : Le problème des réordonnancements**. Ce chapitre s'intéresse au problème des réordonnancements et plus précisément aux différentes approches qui ont été utilisées pour restreindre, de cette manière, l'espace de recherche de la traduction. Nous présentons une étude détaillée sur le système de règles que nous utilisons et le comparons avec les techniques usuelles. Une partie de ce chapitre a été publiée dans (Pécheux *et al.*, 2014) ; une version étendue est publiée dans (Pécheux *et al.*, 2016a).

- **Chapitre 7 : Importance de l'espace des réordonnancements**. Dans ce dernier chapitre, nous évaluons l'importance de la qualité de l'espace des réordonnancements. Nous cherchons à comprendre si un meilleur design peut améliorer les performances globales et si des études plus poussées doivent être menées. Cette étude a été partiellement présentée dans (Pécheux *et al.*, 2014) et complétée dans (Pécheux *et al.*, 2016a).

- **Chapitre 8 : Conclusions**. Nous présentons les conclusions de ce travail dans ce dernier chapitre et évoquons quelques pistes qui nous semblent prometteuses.

Première partie

Modèles exponentiels pour l'apprentissage structuré

Chapitre 2

Modèles exponentiels pour le traitement automatique des langues

Sommaire

Les modèles exponentiels, dits aussi modèles log-linéaires (*log-linear models*) occupent une place importante en traitement automatique des langues, en particulier dans le cadre de l'apprentissage structuré. Dans ce chapitre, nous introduisons les principaux problèmes linguistiques auxquels nous nous sommes intéressés, les notations utilisées tout au long de notre exposé ainsi que les principaux modèles et techniques considérés dans ce manuscrit.

2.1 Le traitement automatique des langues

Le traitement automatique des langues est un domaine à la frontière de nombreuses disciplines (linguistique, informatique, mathématiques, intelligence artificielle) qui étudie la possibilité de confier à des machines la résolution de problèmes ayant trait au langage humain.

La traduction automatique, qui en est donc un sous-domaine, est fortement imprégnée des techniques et des problèmes que l'on trouve de manière plus générale en traitement automatique des langues. De nombreuses étapes de pré-traitement (racinisation, segmentation en unités, reconnaissance d'entités nommées, etc.) sont souvent utilisées pour préparer les corpus nécessaires à l'apprentissage des modèles de traduction. Certaines méthodes de traduction font également usage de catégories grammaticales des mots ou reposent sur une analyse syntaxique préalable. Jurafsky et Martin (2009) proposent une introduction générale et complète à ce domaine.

Parmi de nombreuses approches possibles du traitement automatique des langues, les méthodes statistiques ont permis d'améliorer les performances quantitatives pour de nombreuses tâches, en particulier depuis l'avènement de l'Internet et la possibilité de collecter de grandes quantités de données à moindre coût. On peut appréhender par exemple la traduction automatique comme un problème d'apprentissage statistique où, étant donné une entrée (une séquence de mots), une fonction de prédiction doit renvoyer sa traduction (une autre séquence de mots). De nombreuses approches essaient de ramener des problèmes de prédiction linguistique à des problèmes de classification (par exemple (Li *et al.*, 2004) pour le problème de la lecture robuste), ou à un problème d'étiquetage de séquence (par exemple (Lavergne *et al.*, 2013a) pour le problème de la traduction automatique).

Les méthodes utilisées en apprentissage statistique jouent un rôle important dans l'étude du traitement automatique des langues et en traduction automatique, en particulier, les modèles graphiques (Wainwright et Jordan, 2008; Bishop, 2006), dont les modèles de maximum d'entropie (Ratnaparkhi, 1997) et les champs conditionnels aléatoires (CRF (Lafferty *et al.*, 2001), voir par exemple Sutton et McCallum (2012) pour un exposé détaillé ou (Tellier et Tommasi, 2011) en français) sont des cas particuliers. Certaines techniques algorithmiques sont également

omniprésentes, comme par exemple la programmation dynamique dans différents semi-anneaux (Huang, 2008).

S'il n'est généralement pas possible d'expliciter les règles qui régissent les phénomènes linguistiques, on peut dans certains cas les modéliser approximativement, en utilisant la régularité et les motifs implicites des structures linguistiques sous-jacentes. C'est en tout cas l'hypothèse faite par le cadre de l'apprentissage automatique supervisé qui, à partir d'exemples préalablement annotés, vise à apprendre automatiquement les mécanismes qui conduisent d'un exemple à son annotation. La figure 1.1 propose un exemple de tâche classique de TAL qui peut être traitée de manière efficace par apprentissage automatique. Cette tâche d'analyse morpho-syntaxique, que nous étudions sous divers aspects dans la première partie de ce document, consiste à savoir retrouver pour chaque mot d'une phrase son étiquette morpho-syntaxique. Si l'on dispose d'un corpus annoté, c'est-à-dire des étiquettes de référence, il est possible d'apprendre de manière supervisée un modèle d'étiquetage qui pourra, de manière raisonnable, inférer des séquences d'étiquettes associées à de nouvelles phrases.

Dans le cadre d'un problème de prédiction, on suppose que l'on dispose d'une entrée $x \in \mathcal{X}$ pour laquelle on cherche à prédire la valeur de sortie $y \in \mathcal{Y}$ associée. La détection de spam, la catégorisation grammaticale ou encore la traduction automatique sont des exemples d'instances de ce problème.

Lorsque les espaces $\mathcal{X}$ ou $\mathcal{Y}$ sont complexes, il s'avère souvent nécessaire de prendre en compte leur structure. Le problème de l'étiquetage de séquences consiste à associer à une séquence d'entrée $\mathbf{x} = (x_1, \ldots, x_N)$ une séquence de sortie $\mathbf{y} = (y_1, \ldots, y_N)$, où y_i est la catégorie grammaticale associée au mot x_i. Il est possible de prédire directement une sortie $\mathbf{y}$ pour une entrée $\mathbf{x}$, mais se pose alors un problème combinatoire de taille. Une première possibilité est d'émettre une hypothèse d'indépendance, abusive, sur les composantes y_i de y et de s'intéresser aux sous-problèmes de prédiction de y_i en fonction de x. On préfère cependant utiliser un modèle qui puisse se décomposer suivant la structure de $\mathcal{Y}$, quitte à faire des hypothèses simplificatrices, et tirer profit de cette décomposition à l'aide d'algorithmes adaptés, par exemple de programmation dynamique. On parle d'*apprentissage structuré* lorsque l'on munit $\mathcal{Y}$ d'une structure prise en compte lors du processus de modélisation, ce qui est souvent le cas en traitement automatique des langues. Dans le problème de l'étiquetage de séquence, la structure du modèle provient du fait que l'étiquetage d'une phrase va se décomposer comme l'étiquetage successif des positions d'entrée, avec des hypothèses simplificatrices sur les dépendances entre étiquettes (par exemple Markoviennes).

2.1.1 Analyse morpho-syntaxique

Les catégories morpho-syntaxiques, qui regroupent les mots partageant un même comportement syntaxique et/ou morphologique, constituent une source d'information pertinente pour de nombreuses tâches de TAL. Elles sont par exemple aujourd'hui presque systématiquement calculées en prétraitement pour des tâches d'extraction d'information, pour la reconnaissance d'entités nommées ou encore en traduction automatique, sans parler de leur utilisation en analyse syntaxique. Étant donné leur importance, de nombreux travaux se sont attachés à prédire automatiquement ces étiquettes en utilisant une grande variété de méthodes d'apprentissage supervisé.

Dans le cas de l'analyse morpho-syntaxique, les objets à prédire prennent la forme de séquences et l'espace de sortie est alors un ensemble de séquences. Il est donc naturel de faire appel à des techniques d'apprentissage structuré.

2.1.2 Traduction automatique

Depuis quelques années, la traduction automatique – un problème apparu très tôt en informatique – connaît un essort considérable, principalement en raison de l'avènement d'Internet, tant par la disponibilité d'un nombre croissant de ressources que par les demandes massives et diverses de nombreux utilisateurs dans toutes les langues. Ces progrès considérables ont été rendus possibles par la conjonction de deux événements importants. D'une part, l'avènement d'Internet a massivement contribué à constituer, répertorier et partager de grands volumes de données textuelles, et ce dans plusieurs langues. Il a ainsi été possible de constituer d'importants corpus parallèles, c'est-à-dire de phrases en relation de traduction mutuelle, ce qui fournit un très grand nombre d'exemples. D'autre part, ces dernières décennies ont donné lieu à d'importants développements de méthodes statistiques qui ont peu à peu supplanté les méthodes à base de règles manuelles, peu robustes lorsqu'il s'agit de passer à l'échelle ou au traitement d'énoncés arbitrairement complexes, bruités, ou variés en terme et en registre. Les méthodes statistiques, qui se basent sur le traitement de très grands corpus bilingues, semblent les seules à pouvoir faire face à ces nouvelles exigences.

Cependant, si la traduction automatique a fait d'énormes progrès, les problèmes qu'elle pose sont encore loin d'être résolus, en particulier pour les paires de langues distantes, morphologiquement riches ou pour les langues peu dotées en ressources.

Ces méthodes statistiques utilisent de grandes bases de données (typiquement de très grands corpus bilingues) pour apprendre un certain nombre de paramètres qui constituent le modèle. Ce modèle est ensuite utilisé pour décoder (traduire) de nouvelles phrases. Ceci suppose que l'on dispose d'un corpus bilingue aligné

au niveau des phrases, ce qui, même aujourd'hui, n'est pas le cas pour toutes les paires de langues.

Le problème de la traduction peut se formaliser comme un problème d'apprentissage structuré. À partir d'une phrase source $\mathbf{x} \in \mathcal{X}$, où $\mathcal{X}$ est alors l'ensemble des phrases de la langue source, on cherche à associer « sa » traduction $\mathbf{y} \in \mathcal{Y}$, où $\mathcal{Y}$ est l'ensemble des phrases en langue cible. Cette formulation ne se fait pas sans poser de nombreux problèmes de modélisation que nous examinons par la suite. Si les techniques d'apprentissage statistique sont utilisées à différents niveaux dans les modèles de traduction automatique, peu de travaux ont formalisé la traduction dans son ensemble comme un seul et même modèle structuré, à quelques exceptions près (Zhang *et al.*, 2008; Tillmann et Zhang, 2006; Liang *et al.*, 2006; Blunsom *et al.*, 2008).

Nous présenterons la tâche de traduction automatique et les modèles développés avec davantage de détails au chapitre 5.

2.2 L'apprentissage automatique

La plupart des problèmes trop complexes, et en particulier ceux faisant intervenir les capacités humaines, comme les langues naturelles, la vision ou la reconnaissance d'objets, ne peuvent pas être simplement résolus par des règles algorithmiques. L'*apprentissage statistique*, dit aussi apprentissage automatique, s'intéresse à l'ensemble des méthodes et des modèles qui permettent à une machine d'inférer automatiquement des règles à partir de données pour résoudre de telles tâches. Ces méthodes utilisent des exemples — et sont d'autant plus efficaces que ces exemples sont nombreux — pour collecter des statistiques qui permettent ensuite de choisir de manière probabiliste les meilleures solutions. Pour la tâche d'analyse morpho-syntaxique, à partir de phrases dont on connaît la catégorie morpho-syntaxique de chaque mot, on cherche à développer un modèle permettant de trouver les catégories de nouveaux mots. De même à partir de corpus bilingues de phrases parallèles, on cherche à apprendre automatiquement le procédé qui permette de traduire une phrase d'une langue à une autre.

La théorie de l'apprentissage permet de formaliser le problème — et ainsi de définir ce que l'on entend par « apprentissage » — et, dans certains cas, d'apporter des résultats théoriques qui permettent d'évaluer l'efficacité de différentes méthodes (Amini, 2015). Cette théorie est particulièrement développée pour la classification binaire ou multi-classe. Dans le cas de l'apprentissage structuré, que nous envisageons ici, les bornes théoriques sont souvent complexes à dériver ou trop faibles pour être réellement utilisables en pratique. De plus, les hypothèses principales de l'apprentissage automatique, en particulier le fait de supposer que les

données proviennent d'exemples indépendamment distribués, sont le plus souvent incorrectes en traitement automatique des langues.

2.2.1 Formalisation

On considère un ensemble d'entrée $\mathcal{X}$ et un espace de sortie $\mathcal{Y}$. Comme nous l'avons vu précédemment pour la tâche d'analyse morpho-syntaxique on peut prendre pour $\mathcal{X}$ l'ensemble de toutes les phrases du français et pour $\mathcal{Y}$ l'ensemble des étiquetages morpho-sytaxiques possibles de toutes ces phrases.

Dans le problème de *prédiction*, on cherche à trouver — ce que l'on appellera apprendre — une fonction $f : \mathcal{X} \to \mathcal{Y}$ qui associe à toute entrée $\mathbf{x}$ une sortie $f(\mathbf{x})$ que l'on appelle la prédiction ou l'hypothèse de f. On parle de classification binaire lorsque $|Y| = 2$ et de classification multi-classe lorsque le nombre de sorties $|Y|$ est fini. Le problème de prédiction est alors également appelé problème de *classification*. On parle d'*apprentissage structuré* lorsque l'on munit l'espace de sortie Y d'une structure dont on fait usage lors de la modélisation.

2.2.2 Espace de recherche

Dans certains cas pour une entrée $\mathbf{x}$, l'ensemble des sorties possibles est restreint à un sous-ensemble que l'on note $\mathcal{Y}(\mathbf{x})$. Cela peut relever directement de la formulation du modèle : pour la tâche d'analyse morpho-syntaxique il est inutile de considérer des étiquetages dont la longueur n'est pas celle de la phrase à étiqueter. Si $\mathcal{T}$ désigne l'ensemble des étiquettes morpho-sytaxiques, on peut alors prendre $\mathcal{Y}(\mathbf{x}) = \mathcal{T}^{|\mathbf{x}|}$. En traduction automatique en revanche, bien qu'il semble naturel de ne pas considérer toutes les phrases d'une langue comme candidats de traduction possible, il n'est pas évident *a priori* de définir $\mathcal{Y}(\mathbf{x})$. Dans le chapitre 4, nous étudions l'importance que peut avoir la définition de l'espace de recherche $\mathcal{Y}(\mathbf{x})$, en particulier lorsque l'on cherche à réduire celui-ci. Les chapitres 6 et 7 s'intéressent également à la définition et à l'impact de l'espace de recherche de réordonnancement dans le cas de la traduction automatique.

2.2.3 Fonction de score, inférence

L'étape de modélisation consiste à choisir un ensemble $\mathcal{H}$ de fonctions de prédiction possibles. Dans l'approche *paramétrique* on spécifie la forme du modèle $\mathcal{H} = \{f_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta}$ par un ensemble de paramètres $\Theta \subset \mathbb{R}^d, d \geq 1$.

Une approche classique en apprentissage structuré est de définir une fonction de score $s_\theta : \mathcal{X} \to \mathcal{Y}$ dépendant d'un paramètre $\theta \in \Theta$. On définit alors la fonction de prédiction par

$$\forall \mathbf{x} \in \mathcal{X}, f_{\boldsymbol{\theta}}(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}(\mathbf{x})} s_{\theta}(\mathbf{x}, \mathbf{y}) \tag{2.1}$$

On parle alors également d'*inférence* ou de *décodage* comme synonyme de prédiction, c'est-à-dire lors du calcul de l'équation (2.1), cette étape n'étant généralement pas triviale.

2.2.4 Modèles linéaires et caractéristiques

En TAL, on considère le plus souvent des modèles *linéaires* $s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$, paramétrés par $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^d$, où $\boldsymbol{\phi} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ est une fonction vectorielle. Pour $\mathbf{x} \in \mathcal{X}$ et $\mathbf{y} \in \mathcal{Y}$ on dit que $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$ est le *vecteur de caractéristiques* de $(\mathbf{x}, \mathbf{y})$. Le modèle est dit linéaire en raison de sa linéarité vis-à-vis des paramètres $\boldsymbol{\theta}$. De manière plus générale, il suffit que les frontières de décision de l'équation (2.1) soient linéaires pour que le modèle soit qualifié de linéaire.

Les modèles linéaires sont ainsi fondés sur une combinaison linéaire de caractéristiques. Une caractéristique, c'est-à-dire une composante k du vecteur de caractéristiques, associe à chaque couple d'entrée sortie $(\mathbf{x}, \mathbf{y})$ une valeur réelle $\boldsymbol{\phi}_k(\mathbf{x}, \mathbf{y}) \in \mathbb{R}$, par exemple une propriété, une mesure de compatibilité, une distance, etc. On utilise souvent des caractéristiques binaires, à valeur dans $\{0, 1\}$ qui sont *actives*, c'est-à-dire prennent la valeur 1 lorsqu'une certaine propriété de $(\mathbf{x}, \mathbf{y})$ est vérifiée. Par exemple, si $\mathbf{x}$ et $\mathbf{y}$ sont des phrases, on peut avoir une caractéristique qui est active si et seulement si $\mathbf{x}$ et $\mathbf{y}$ ont le même nombre de mots, commencent toutes les deux par une majuscule, sont « syntaxiquement correctes » ou encore présentent une combinaison de ces dernières propriétés.

2.2.5 Apprentissage

Dans les modèles linéaires de la section 2.2.4, un poids est associé à chaque caractéristique pour en former une combinaison linéaire. La valeur de chaque poids oriente la décision finale de l'équation (2.1) et on souhaiterait donc les choisir de manière à ce que la fonction de prédiction $f_{\boldsymbol{\theta}}$ soit la « meilleure » possible.

L'apprentissage statistique consiste justement à calculer — on dit aussi « apprendre » ou estimer — les paramètres $\boldsymbol{\theta} \in \Theta$ à partir de certaines observations dans le but précis de satisfaire cet objectif, c'est-à-dire d'optimiser les prédictions. L'apprentissage revient donc à choisir dans la famille de modèles paramétrés $\mathcal{H} = \{f_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta}$ le modèle $f_{\boldsymbol{\theta}}$ qui convient le mieux.

Dans le cadre de l'*apprentissage supervisé* on dispose d'un ensemble $\mathcal{D} = \{(\tilde{\mathbf{x}}^{(i)}, \tilde{\mathbf{y}}^{(i)})\}_{i=1}^{N}$ de N exemples de référence. Le cadre théorique de l'apprentissage statistique suppose l'existence d'une loi $p^*(\mathbf{x}, \mathbf{y})$ qui représente la « vraie » distribution du problème que l'on cherche à modéliser. Par exemple $p^*(\mathbf{x}, \mathbf{y})$ peut désigner

la probabilité qu'une certaine phrase $\mathbf{x}$ soit prononcée par un locuteur d'une certaine langue source et qu'un interprète la traduise par la phrase en langue cible $\mathbf{y}$. On suppose en théorie que $\mathcal{D}$ est un échantillon identiquement et indépendamment distribué (i.i.d) suivant la loi p^*. Cette loi est difficile à caractériser et varie de manière non contrôlée. De ce fait, en TAL, cette hypothèse n'est jamais vérifiée en pratique. La constitution de corpus bilingues, qui permet d'obtenir les données $\mathcal{D}$, obéit à diverses contraintes, filtrages, biais, et les doublons sont souvent supprimés des corpus d'apprentissage.

La théorie de l'apprentissage supervisé est souvent formulée dans le cadre de la minimisation du risque empirique.

2.2.6 Cadre de la minimisation du risque empirique

Le cadre de la minimisation du risque empirique propose un formalisme dans lequel il est possible de représenter et de comparer de nombreuses approches d'apprentissage statistique. La formalisation fait appel à une fonction de *perte*

$$perte : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathbb{R}_+$$

dont on peut donner l'interprétation suivante :

> $perte(x, y; f)$ est la perte encourue lorsque l'entrée est x, la fonction de prédiction f et la sortie de référence y.

Exemple 2.2.1. *Une des pertes les plus usuelles, dite « tout ou rien » ou* perte 0-1 *vaut* 0 *si la prédiction est correcte et* 1 *sinon :* $perte_{0-1}(x, y; f) = \mathbb{1}_{\{f(x) \neq y\}}$.

On cherche ensuite à minimiser la valeur prise par cette fonction sur les exemples tirés de p^*, en général en considérant les exemples de $\mathcal{D}$.

Formellement, on cherche la fonction de prédiction $f \in \mathcal{H}$ qui minimise le risque $\mathcal{R}(f) = \mathbb{E}_{(x,y)\sim p^*}[perte(x, y; h)]$:

$$f^* = \min_{f \in \mathcal{H}} \{\mathcal{R}(f)\} \tag{2.2}$$

Comme on ne connaît pas la probabilité p^* sous-jacente mais que l'on a accès à un échantillon de données $\mathcal{D} = \{(\tilde{\mathbf{x}}^{(i)}, \tilde{\mathbf{y}}^{(i)})\}_{i=1}^N$, on approche le risque par le *risque empirique* calculé en utilisant la probabilité empirique $\hat{p}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{\{(\mathbf{x},\mathbf{y})=(\tilde{\mathbf{x}}^{(i)},\tilde{\mathbf{y}}^{(i)})\}}$. En pratique, on considère plutôt le *risque empirique régularisé* en ajoutant une mesure quantifiant la « complexité » de la fonction de prédiction :

$$\hat{\mathcal{R}}(f) = \mathbb{E}_{(x,y)\sim\hat{p}}[perte(x, y; f)] + \text{complexité}(f) \tag{2.3}$$

$$= \frac{1}{N}\sum_{i=1}^{N} perte(x_i, y_i; f) + \text{complexité}(f) \tag{2.4}$$

Dans le cas des modèles paramétrés cela donne :

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} perte(\tilde{\mathbf{x}}^{(i)}, \tilde{\mathbf{y}}^{(i)}, \boldsymbol{\theta}) + R(\boldsymbol{\theta}) \tag{2.5}$$

La mesure de complexité, que l'on appelle aussi facteur de *régularisation*, notée $R(\boldsymbol{\theta})$ pour les modèles paramétrés, permet de privilégier les modèles plus simples et ainsi d'éviter d'apprendre des particularités spécifiques à l'échantillon considéré.

Un des intérêts de ce formalisme — en dehors du fait que ce cadre théorique est relativement bien étudié et permet dans de nombreux cas de donner des garanties théoriques sous des hypothèses idoines, est qu'il permet de formaliser, d'analyser et de comparer de nombreuses approches selon la manière dont elles envisagent et définissent la fonction de perte. Nous présenterons différentes fonctions de perte plus loin : le maximum de vraisemblance dans un premier temps (§ 2.3.3), puis d'autres fonctions de pertes possibles dans la section 2.4.

Une fonction de perte simple est celle présentée dans l'exemple 2.2.1 qui revient à compter combien d'exemples du corpus d'apprentissage sont correctement prédits. Cette fonction de perte pose cependant deux problèmes principaux : d'une part le problème d'optimisation qui en résulte n'est en général pas convexe et son optimisation est compliquée ; d'autre part l'utilisation d'une mesure « tout ou rien » n'est pas toujours adaptée aux problèmes de TAL, comme nous allons en discuter à la section suivante.

2.2.7 Mesures d'évaluation et fonctions de coût

Lorsque l'on définit une tâche de TAL, il est utile d'avoir une mesure d'évaluation pour quantifier la qualité des solutions trouvées. Pour la tâche d'analyse morpho-syntaxique, on peut considérer qu'une prédiction est juste si et seulement si toutes les étiquettes correctes ont été trouvées. On peut également choisir une mesure moins sévère, proportionnelle au nombre d'étiquettes correctes, choix qui est fait habituellement. Dans certains cas, le choix d'une métrique d'évaluation ne va pas de soi. Par exemple, en traduction automatique, il n'existe pas de métrique automatique qui soit capable de rendre compte d'une notion de qualité conforme au jugement humain. Différents choix sont possibles, comme nous l'exposerons à

la section 5.5, la plupart reposant sur des concordances partielles entre l'hypothèse et la ou les traductions de référence.

Une fois fixée une mesure d'évaluation, il semble intéressant de prendre celle-ci en compte, directement ou indirectement, lors du processus d'apprentissage. On formalise généralement cela par l'introduction d'une fonction de *coût* : $\mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ où $coût(\hat{\mathbf{y}}, \mathbf{y})$ quantifie le coût à prédire $\mathbf{y}$, lorsque la bonne prédiction [1] était $\hat{\mathbf{y}}$.

Une des métriques les plus utilisées en traduction est BLEU dont nous parlerons dans la section 5.5, qui donne une mesure de qualité entre une référence $\tilde{\mathbf{y}}$ et une hypothèse $\mathbf{y}$ à valeur dans $[\![0, 1]\!]$. On peut alors prendre comme fonction de coût $1 - BLEU(\tilde{\mathbf{y}}, \mathbf{y})$.

Nous reviendrons sur l'intégration du coût dans la fonction de perte dans la section 2.4.

2.2.8 Approches génératives et discriminantes

En apprentissage statistique — et en particulier lorsque l'on applique celui-ci au TAL, on oppose souvent deux approches : les méthodes dites *génératives* d'une part et les méthodes *discriminantes* d'autre part (Wainwright et Jordan, 2008; Smith, 2011). Dans de nombreux domaines, on a peu à peu cherché à remplacer les modèles génératifs en usage par de nouvelles méthodes discriminantes. La présentation qui suit et de la section 2.3 est inspirée des travaux de Bouchard et Triggs (2004); Bouchard (2007); Lasserre *et al.* (2006); Minka (2005) et Altun *et al.* (2003).

La comparaison des deux approches est étudiée depuis longtemps d'un point de vue théorique, principalement par des études asymptotiques (Efron, 1975; Liang et Jordan, 2008), mais Ng et Jordan (2002) ont réalisé une étude non asymptotique.

2.2.8.1 Approches génératives

Le mot *génératif* est un terme générique pour identifier les approches qui définissent un modèle probabiliste sur la distribution jointe $p(\mathbf{x}, \mathbf{y})$, dit *modèle génératif*. On modélise ainsi toutes les variables et les contraintes sur la structure de $\mathcal{X} \times \mathcal{Y}$ sont encodées par ces probabilités jointes :

$$\mathcal{G} = \{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}), \theta \in \Theta_G\} \tag{2.6}$$

Le plus souvent on factorise et on estime cette probabilité jointe par une probabilité conditionnelle de classe et une probabilité *a priori* sur la classe : $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$. Un classifieur naturel utilise alors la règle de Bayes pour choisir la classe de plus grande probabilité *a posteriori* $p(\mathbf{y}|\mathbf{x})$.

1. Remarquons que l'entrée $\mathbf{x}$ est ici ignorée. On peut étendre ce cadre lorsque plusieurs solutions de référence existent.

Dans le milieu biomédical, cette approche est parfois appelée le *paradigme d'échantillonnage*. En effet, les modèles génératifs définissent souvent une *histoire générative*, c'est-à-dire une description de la manière dont ont été virtuellement engendrées les données.

Un des avantages des modèles génératifs est qu'ils permettent de faire de *l'inférence probabiliste* et de répondre à de nombreuses questions que l'on peut se poser sur certains sous-ensembles des variables. Cependant, si l'on se restreint au cadre purement prédictif où $\mathbf{x}$ est toujours observé, on peut se demander quelle est l'utilité de modéliser $\mathbf{x}$. Cela peut même au contraire poser problème : supposons que l'on observe un nouvel exemple $(\mathbf{x}, \mathbf{y})$ qui est bien classé par notre modèle, mais dans lequel $\mathbf{x}$ n'est pas très bien expliqué. Même si notre classifieur donne la bonne réponse, il faut remettre en cause tout le modèle pour expliquer cette nouvelle information. Les méthodes discriminantes introduites à la section suivante répondent à ce problème.

2.2.8.2 Approches discriminantes

D'une certaine manière, on peut dire que le terme *discriminant* ne se définit vraiment que par opposition au terme *génératif*, distinction qui n'a de sens que pour des modèles probabilistes (Wainwright et Jordan, 2008). Dans apprentissage discriminant il y a aussi l'idée d'optimiser une fonction de perte liée à la mesure de performance, alors que l'apprentissage génératif est davantage liée à l'estimation de densité.

Les méthodes probabilistes discriminantes s'intéressent et modélisent directement la probabilité conditionnelle $p(\mathbf{y}|\mathbf{x})$ et considèrent donc une famille :

$$\mathcal{D} = \{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}), \theta \in \Theta_D\}$$

Dans de nombreuses applications $\mathcal{X}$ est un espace de très grande dimension et $\mathbf{x}$ est difficile à modéliser ou non aléatoire (il serait alors problématique d'utiliser l'hypothèse i.i.d.). Les modèles conditionnels ne modélisent pas l'espace $\mathcal{X}$. Si le but final est de trouver la règle de classification d'erreur minimale qui ne dépend que de $p(\mathbf{y}|\mathbf{x})$, il est inutile de résoudre comme problème intermédiaire un problème plus général, et les approches discriminantes ne font donc pas plus de travail que nécessaire.

L'approche discriminante se concentre directement sur la séparation des classes, sans chercher à les modéliser, ce qui présente un avantage si cette modélisation est délicate. Cette approche est d'ailleurs appelée le *paradigme diagnostique* dans le monde biomédical.

La différence entre l'apprentissage génératif et l'apprentissage discriminant est aussi une question de point de vue, et de l'histoire que l'on cherche à raconter. Dans

l'apprentissage génératif, on cherche à modéliser le processus, à le décomposer en sous-étapes indépendantes et à estimer les paramètres de cette histoire générative. Dans l'apprentissage discriminant, on cherche plutôt à optimiser directement les performances d'une mesure d'erreur.

2.3 Modèles exponentiels

Dans cette partie on s'intéresse plus particulièrement aux modèles probabilistes, en particulier les modèles exponentiels, et à l'apprentissage par maximum de vraisemblance.

2.3.1 Modèles probabilistes

Il existe de nombreuses manières intéressantes de choisir la fonction de score. Dans cette section on s'intéresse aux modèles probabilistes, pour lesquels la fonction de score s_θ est également une fonction de probabilité. On peut distinguer deux types de modèles, les modèles probabilistes définissant une distribution jointe, que l'on appelle souvent modèles *génératifs*, et ceux définissant une probabilité conditionnelle, que l'on dit également *discriminants*.

Modèle conditionnel $\forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{y} \in \mathcal{Y}(\mathbf{x}), \quad s_\theta(\mathbf{x}, \mathbf{y}) = p_\theta(\mathbf{y}|\mathbf{x})$
où $\forall \mathbf{x} \in \mathcal{X}$, $p_\theta(.|\mathbf{x}) : \mathcal{Y}(\mathbf{x}) \to \mathbb{R}$ est une fonction de probabilité, *i.e.* vérifiant :

$$\forall \mathbf{y} \in \mathcal{Y}(\mathbf{x}), p_\theta(\mathbf{y}|\mathbf{x}) \geq 0$$

$$\sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} p_\theta(\mathbf{y}|\mathbf{x}) = 1$$

Remarque 2.3.1. *On modélise ici explicitement, par rappport au paramètre θ, uniquement la probabilité conditionnelle $p_\theta(\mathbf{y}|\mathbf{x})$. On peut également imaginer que l'on modélise, arbitrairement et implicitement, la probabilité sur $\mathbf{x}$, $p_{\theta'}(\mathbf{x})$ mais à l'aide d'un autre paramètre $\theta' \in \mathbb{R}^{d'}$ indépendant et aussi expressif que l'on souhaite et que l'on ne cherchera pas à apprendre (Minka, 2005; Lasserre et al., 2006).*

Modèle joint $\forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{y} \in \mathcal{Y}(\mathbf{x}), \quad s_\theta(\mathbf{x}, \mathbf{y}) = p_\theta(\mathbf{x}, \mathbf{y})$
où $p_\theta : \{(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}(\mathbf{x})\} \to \mathbb{R}$ est une fonction de probabilité, *i.e.* vérifiant :

$$\forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{y} \in \mathcal{Y}(\mathbf{x}), p_\theta(\mathbf{x}, \mathbf{y}) \geq 0$$

$$\sum_{\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}(\mathbf{x})} p_\theta(\mathbf{x}, \mathbf{y}) = 1$$

Remarque 2.3.2. *On modélise ici explicitement $p_\theta(\mathbf{x}, \mathbf{y})$, que l'on peut décomposer en $p_\theta(\mathbf{y}|\mathbf{x})p_\theta(\mathbf{x})$. Contrairement à l'approche précédente, ici $\theta = \theta'$, et l'on a d'une certaine manière une moins grande marge de liberté de modélisation par rapport à l'approche précédente où $\boldsymbol{\theta}'$ pouvait être quelconque. On pourrait cependant ici aussi, lors de la modélisation, décomposer $\boldsymbol{\theta}$ de manière à séparer sa contribution dans $p_\theta(\mathbf{y}|\mathbf{x})$ et dans $p_\theta(\mathbf{x})$ en deux partitions indépendantes. Ce qui diffère donc réellement par rapport au modèle conditionnel, comme nous le verrons par la suite, c'est que l'on va chercher à apprendre conjointement ce (ou ces) paramètre(s).*

Conclusion Dans un modèle conditionnel on ne modélise pas $p_{\theta'}(\mathbf{x})$. À partir d'un modèle joint, on peut directement associer un modèle conditionnel correspondant. Comme on peut passer simplement de l'un à l'autre par la règle de Bayes, on ne peut pas directement dire de la paramétrisation d'un modèle si elle est jointe ou conditionnelle. Ce qui va importer c'est la manière dont sont appris les paramètres.

Remarque 2.3.3. *Klein et Manning (2002) distinguent, parmi les modèles joints, ceux à* structure *générative où l'on décompose la probabilité jointe sous la forme $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ et les modèles à* structure *conditionnelle où on décompose la probabilité jointe sous la forme $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x})$. Dans ce dernier cas, ils ignorent ensuite le terme $p(\mathbf{x})$ lors de l'apprentissage, ce qui revient donc plutôt à un modèle conditionnel tel que présenté ci-dessus.*

2.3.2 Inférence

Remarquons que $\forall \mathbf{x} \in \mathcal{X}$

$$\begin{aligned} f(\mathbf{x}) &= \arg\max_{y \in \mathcal{Y}(\mathbf{x})} p_\theta(\mathbf{x}, \mathbf{y}) \\ &= \arg\max_{y \in \mathcal{Y}(\mathbf{x})} p_\theta(\mathbf{y}|\mathbf{x}) p_\theta(\mathbf{x}) \\ &= \arg\max_{y \in \mathcal{Y}(\mathbf{x})} p_\theta(\mathbf{y}|\mathbf{x}) \end{aligned}$$

Inversement, à partir de la probabilité conditionnelle, quitte à multiplier par exemple[2] par $p_{\theta'}(\mathbf{x}) = \frac{1}{|\mathcal{X}|}$ on retrouve une probabilité jointe qui n'a aucune incidence sur la fonction de prédiction.

Remarquons qu'ici peu importe que les paramètres $\boldsymbol{\theta}$ définissent une probabilité et que celle-ci soit conditionnelle ou non. La seule différence va se situer au

2. Mais on peut prendre n'importe quelle probabilité $p_{\theta'}(\mathbf{x})$, en particulier si $\mathcal{X}$ n'est pas un ensemble fini.

niveau de l'apprentissage, c'est-à-dire dans la manière dont est choisi le vecteur de paramètres $\boldsymbol{\theta}$.

À l'inférence donc, le choix d'une fonction de score comme probabilité jointe ou conditionnelle ne change rien, seule la partie conditionnelle $p_\theta(\mathbf{y}|\mathbf{x})$ est utilisée. C'est une des raisons principales invoquées pour motiver une modélisation conditionnelle : pourquoi alors modéliser l'espace d'entrée $\mathcal{X}$ alors que cela n'est pas utilisé par la fonction de prédiction [3] ?

2.3.3 Apprentissage par maximum de vraisemblance

Au vu de la partie précédente, le but de l'apprentissage est de trouver un paramètre θ qui définisse un « bon » modèle $p_\theta(\mathbf{y}|\mathbf{x})$.

Une approche classique est celle du maximum de vraisemblance, qui présente des propriétés théoriques intéressantes. Dans cette approche on choisit θ^* en maximisant une fonction de vraisemblance $\mathcal{L}_\theta(\mathcal{D})$ des paramètres au vu des données

$$\theta^* = \arg\max_\theta \mathcal{L}_\theta(\mathcal{D}) \tag{2.7}$$

On peut encore plus ou moins distinguer une approche conditionnelle (associée à la notion d'apprentissage *discriminant*) d'une approche jointe (associée à la notion d'apprentissage *génératif*) en fonction de la manière dont on définit la log-vraisemblance, traduisant le modèle que l'on utilise pour la définir. Remarquons encore qu'une modélisation jointe $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})$ induit une probabilité conditionnelle $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})}{\sum_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})}$, et peut donc être entraînée de deux manières.

Apprentissage conditionnel On définit la log-vraisemblance conditionnelle par

$$\mathcal{L}^C_\theta(\mathcal{D}) = \sum_{i=1}^{N} \log p_\theta(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) \tag{2.8}$$

On a alors

$$\theta^* = \arg\max_\theta \sum_{i=1}^{N} \log p_\theta(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) \tag{2.9}$$

$$= \arg\max_\theta \sum_{i=1}^{N} \log p_\theta(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) + \log p_{\theta'}(\mathbf{x}^{(i)}) \quad \forall \theta' \in \mathbb{R}^{d'} \tag{2.10}$$

3. Cela peut cependant avoir un intérêt dans un cadre plus large que celui de la simple prédiction considérée ici.

L'équation 2.10 montre simplement que formellement la probabilité implicite (non-modélisée) sur $\mathcal{X}$ peut-être ajoutée mais n'est pas apprise.

Apprentissage joint On définit la log-vraisemblance jointe par

$$\mathcal{L}_\theta^J(\mathcal{D}) = \sum_{i=1}^{N} \log p_\theta(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$$

On a alors

$$\theta^* = \arg\max_\theta \sum_{i=1}^{N} \log p_\theta(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \tag{2.11}$$

$$= \arg\max_\theta \sum_{i=1}^{N} \log p_\theta(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) + \log p_\theta(\mathbf{x}^{(i)}) \tag{2.12}$$

$$= \arg\max_\theta \sum_{i=1}^{N} \log p_\theta(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) + \mathcal{R}(\mathbf{x}^{(i)}, \theta) \tag{2.13}$$

L'équation 2.13, par contraste avec l'équation 2.9, montre que la différence entre les deux approches peut s'interpréter comme l'ajout dans le cas génératif d'un terme $\mathcal{R}(\mathbf{x}^{(i)}, \theta) = \log p_\theta(\mathbf{x}^{(i)})$ que l'on peut assimiler à une forme de régularisation [4] (Sutton et McCallum, 2012).

Le problème de l'apprentissage joint est qu'en pratique il n'est pas facile de définir des modèles joints pour lesquels l'inférence ou l'apprentissage soit possible en temps raisonnable, à moins de faire un certain nombre d'hypothèses. De nombreuses hypothèses d'indépendance sont ainsi nécessaires, hypothèses qui ont également un impact sur la forme que peut prendre $p_\theta(\mathbf{y}|\mathbf{x})$. Une des raisons du succès des approches probabilistes discriminantes est de permettre une plus grande liberté dans le choix de $p_\theta(\mathbf{y}|\mathbf{x})$.

La différence entre une modélisation jointe et une modélisation conditionnelle réside dans le couplage qui est fait entre entre θ et θ' dans l'équation (2.10). Le modèle joint impose $\theta = \theta'$ alors que le modèle conditionnel ne pose aucune contrainte sur θ', ce qui conduit Minka (2005) à conclure que l'approche discriminante est plus souple. Cela dépend en réalité de la manière dont $p_{\boldsymbol{\theta}}(\mathbf{x})$ et $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ dépendent précisément de θ et à l'extrême il est toujours possible de découpler les deux. On comprend qu'il y a un continuum entre les deux approches, qui est exploité

4. Généralement, les modèles joints autant que conditionnels sont également régularisés par ailleurs, par exemple par un terme $\mathcal{R}(\theta) = -\lambda\|\theta\|_2^2$, en complément aux équations (2.9) et (2.11). Ce terme, assimilable à une forme de régularisation qui distingue les deux modèles, est complémentaire.

par (Lasserre *et al.*, 2006) pour réaliser des modèles hybrides, entre génératifs et discriminants.

Klein et Manning (2002) observent, de manière empirique sur diverses tâches de traitement des langues naturelles, que les modèles joints ont de meilleures performances que les modèles à structure conditionnelle. Parmi les modèles joints, ils observent également que l'entraînement discriminant (i.e. conditionnel) est légèrement préférable. Roark *et al.* (2004) observent cependant le comportement contraire. Altun *et al.* (2003) comparent différents critères d'apprentissage et montrent que plus que la méthode d'apprentissage, c'est la capacité d'un modèle à utiliser des caractéristiques plus riches qui permet d'améliorer sensiblement les différences, ce qui est davantage possible dans l'approche conditionnelle.

Lien avec la divergence de Kullback-Leibler Une autre vision possible consiste à adopter une approche probabiliste et à estimer les paramètres $\boldsymbol{\theta}$ de manière à ce que $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$, resp. $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})$, soit le plus « proche possible » de la « vraie » distribution $p^*(\mathbf{y}|\mathbf{x})$, resp. $p^*(\mathbf{x},\mathbf{y})$. Ces distributions étant inconnues, on les approche par les probabilités empiriques $\hat{p}(\mathbf{y}|\mathbf{x})$, resp. $\hat{p}(\mathbf{x},\mathbf{y})$. Comme mesure de proximité, il est courant d'utiliser la divergence de Kullback-Leibler $D(p,q) = \sum_z p(z) \log \frac{p(z)}{q(z)}$. Il est intéressant de voir que cette approche revient alors au même que l'approche du maximum de vraisemblance.

Apprentissage non-supervisé Lorsque l'on ne dispose pas de sorties annotées, il est possible de considérer le modèle joint et de chercher à maximiser la vraisemblance des données observées :

$$\mathcal{L}_\theta^{NS}(\mathcal{D}) = \sum_{i=1}^{N} \log p_\theta(\mathbf{x}^{(i)}) \tag{2.14}$$

$$= \sum_{i=1}^{N} \log \sum_{\mathbf{y}\in\mathcal{Y}(\mathbf{x}^{(i)})} p_\theta(\mathbf{x}^{(i)},\mathbf{y}) \tag{2.15}$$

2.3.4 Les modèles conditionnels aléatoires

On considère dans cette partie des modèles exponentiels, dits aussi log-linéaires [5] en partant d'une modélisation jointe dont la forme est la suivante

$$\forall \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}(\mathbf{x}), \quad p_\theta(\mathbf{x},\mathbf{y}) = \frac{1}{\mathcal{Z}_\theta} \exp(\theta^T \phi(\mathbf{x},\mathbf{y})) \tag{2.16}$$

5. Cette dénomination vient du fait qu'en prenant le logarithme on obtient une fonction linéaire des paramètres. Ces modèles sont cependant linéaires au sens donné à la section 2.2.4 et nous préférerons donc la dénomination « modèles exponentiels » qui prête moins à confusion.

où $\phi(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^d$ est le vecteur de caractéristiques et

$$\mathcal{Z}_\theta = \sum_{\mathbf{x}\in\mathcal{X}, \mathbf{y}\in\mathcal{Y}(x)} \exp(\theta^T \phi(\mathbf{x}, \mathbf{y})) \tag{2.17}$$

la constante de normalisation.

Le modèle conditionnel correspondant est

$$\forall \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}(\mathbf{x}), \quad p_\theta(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathcal{Z}_\theta(\mathbf{x})} \exp(\theta^T \phi(\mathbf{x}, \mathbf{y})) \tag{2.18}$$

où la constante de normalisation est maintenant

$$\mathcal{Z}_\theta(\mathbf{x}) = \sum_{\mathbf{y}\in\mathcal{Y}(x)} \exp(\theta^T \phi(\mathbf{x}, \mathbf{y})) \tag{2.19}$$

Notons que

$$p_\theta(\mathbf{x}) = \frac{\mathcal{Z}_\theta(\mathbf{x})}{\mathcal{Z}_\theta} \tag{2.20}$$

Le vecteur de caractéristiques $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$ peut en théorie porter sur n'importe quel aspect de l'entrée $\mathbf{x}$ et de la sortie $\mathbf{y}$. En général, la taille des espaces $\mathcal{X}$ et $\mathcal{Y}$ ne permet pas d'énumérer leurs éléments et une forme de structure est nécessaire pour pouvoir calculer les sommes (2.19) et (2.17), en permettant de factoriser les calculs, à condition que les caractéristiques suivent cette structure. L'avantage du modèle conditionnel est de ne contraindre les caractéristiques que sur les sorties $\mathbf{y}$ pour calculer la somme (2.19) alors que le modèle joint nécessite de limiter les caractéristiques possibles sur $\mathcal{X}$ et sur $\mathcal{Y}$.

Pour reprendre les termes ci-dessus, on définit donc un modèle joint que l'on peut entraîner de manière jointe ou de manière conditionnelle.

On a alors

$$\mathcal{L}_\theta^J(\mathcal{D}) = \sum_{i=1}^N \log p_\theta(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \tag{2.21}$$

$$= \sum_{i=1}^N \theta^T \phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \log \mathcal{Z}_\theta \tag{2.22}$$

$$= \underbrace{\sum_{i=1}^N (\theta^T \phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \log \mathcal{Z}_\theta(\mathbf{x}^{(i)})}_{\mathcal{L}_\theta^C(\mathcal{D})} + \overbrace{\log \mathcal{Z}_\theta(\mathbf{x}^{(i)}) - \log \mathcal{Z}_\theta)}^{\mathcal{R}(\mathbf{x}^{(i)}, \theta)} \tag{2.23}$$

La plupart des approches pour optimiser l'équation 2.11 nécessitent le calcul du gradient de $\mathcal{L}_\theta^J(\mathcal{D})$. Les gradients se calculent facilement en remarquant que

$$\nabla_{\boldsymbol{\theta}} \log \mathcal{Z}_{\boldsymbol{\theta}} = \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})} \left[\boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \right] \tag{2.24}$$

Démonstration.

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \log \mathcal{Z}_{\boldsymbol{\theta}} &= \frac{\nabla_{\boldsymbol{\theta}} \mathcal{Z}_{\boldsymbol{\theta}}}{\mathcal{Z}_{\boldsymbol{\theta}}} \\
&= \frac{\nabla_{\boldsymbol{\theta}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{X} \times \mathcal{Y}(\mathbf{x})} \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})\right)}{\mathcal{Z}_{\boldsymbol{\theta}}} \\
&= \frac{\sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{X} \times \mathcal{Y}(\mathbf{x})} \nabla_{\boldsymbol{\theta}} \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})\right)}{\mathcal{Z}_{\boldsymbol{\theta}}} \\
&= \frac{\sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{X} \times \mathcal{Y}(\mathbf{x})} \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})\right)}{\mathcal{Z}_{\boldsymbol{\theta}}} \\
&= \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{X} \times \mathcal{Y}(\mathbf{x})} \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})
\end{aligned}$$

□

et de même

$$\nabla_{\boldsymbol{\theta}} \log \mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})} \left[\boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \right] \tag{2.25}$$

Démonstration.

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \log \mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x}) &= \frac{\nabla_{\boldsymbol{\theta}} \mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x})}{\mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x})} \\
&= \frac{\nabla_{\boldsymbol{\theta}} \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})\right)}{\mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x})} \\
&= \frac{\sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \nabla_{\boldsymbol{\theta}} \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})\right)}{\mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x})} \\
&= \frac{\sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})\right)}{\mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x})} \\
&= \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})
\end{aligned}$$

□

On a alors

$$\nabla_\theta \mathcal{L}_\theta^J(\mathcal{D}) = \underbrace{\sum_{i=1}^{N}(\phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \mathbb{E}_{p_\theta(\mathbf{y}|\mathbf{x}^{(i)})}\left[\phi(\mathbf{x}^{(i)}, \mathbf{y})\right]}_{\nabla_\theta \mathcal{L}_\theta^C(\mathcal{D})} - \mathbb{E}_{p_\theta(\mathbf{y}|\mathbf{x}^{(i)})}\left[\phi(\mathbf{x}^{(i)}, \mathbf{y})\right] + \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{y})}\left[\phi(\mathbf{x}, \mathbf{y})\right]) \tag{2.26}$$

Pour entraîner un modèle de manière générative, il faut donc savoir calculer $\mathbb{E}_{p_\theta(\mathbf{x},\mathbf{y})}\left[\phi(\mathbf{x}, \mathbf{y})\right]$. Dans le cas d'un entraînement discriminant on ne calcule que les deux premiers termes de l'équation 2.26 et il suffit de savoir calculer $\mathbb{E}_{p_\theta(\mathbf{y}|\mathbf{x})}\left[\phi(\mathbf{x}, \mathbf{y})\right]$ ce qui est généralement moins coûteux. Pour pouvoir calculer efficacement ces espérances de manière exacte [6], on est conduit à faire des hypothèses d'indépendance entre les variables. L'approche discriminante permet ici de ne faire aucune hypothèse particulière sur $\mathcal{X}$.

2.3.5 Chaîne linéaire simple

Le calcul de $\mathbb{E}_{p_\theta(\mathbf{x},\mathbf{y})}\left[\phi(\mathbf{x}, \mathbf{y})\right]$ n'est en général possible que si la structure du modèle graphique sous-jacente est un arbre (Wainwright et Jordan, 2008). On considère dans cette partie le cas où les éléments des espaces $\mathcal{X}$ et $\mathcal{Y}$ sont structurés sous la forme de chaînes linéaires. Pour simplifier, on considère que toutes les chaînes ont la même longueur K. On a alors $\mathcal{X} = V^K$ où V est un ensemble fini appelé vocabulaire des mots, $\mathcal{Y} = \{y_0\} \times \mathcal{T}^K$ où $\mathcal{T}$ est un ensemble fini d'étiquettes possibles et $y_0 \notin \mathcal{T}$ un symbole initial. On note $\mathbf{x} = (x_1, \ldots, x_K)$ et $\mathbf{y} = (y_0, y_1, \ldots, y_K)$. Pour chaque $\mathbf{x} \in \mathcal{X}'$, $\mathcal{Y}(\mathbf{x}) = \mathcal{Y}(x_1) \times \mathcal{Y}(x_2) \times \cdots \times \mathcal{Y}(x_K) \subset \mathcal{Y}$.

HMM Considérons la représentation graphique en chaîne linéaire de la colonne centrale, en bas sur la figure 2.1 sous la forme de graphe de facteurs. On peut alors factoriser la probabilité jointe

6. Il existe des méthodes d'inférence approchée que nous n'étudions pas ici.

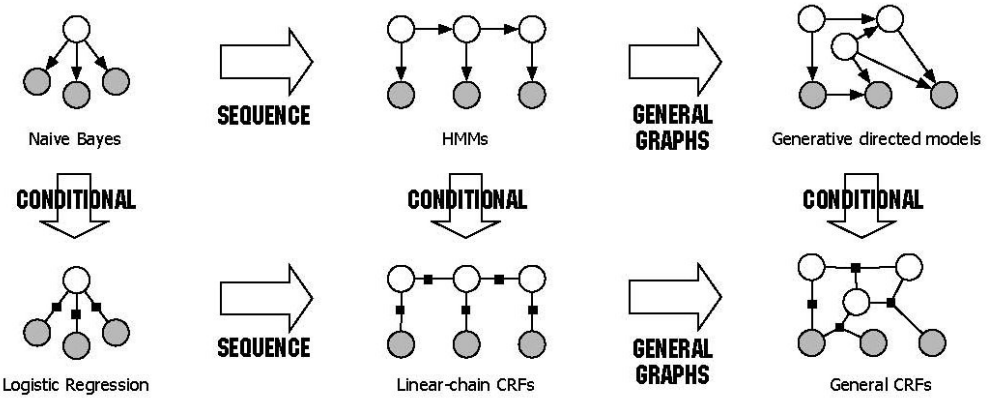


FIGURE 2.1 – Diagramme des relations entre les modèles Naïve Bayes, régression logistique, HMM, CRF en chaîne linéaire, modèles génératifs et CRF (figure reprise de (Sutton et McCallum, 2012)).

$$p_\theta(\mathbf{x},\mathbf{y}) = \frac{1}{\mathcal{Z}_\theta}\exp(\theta^T\phi(\mathbf{x},\mathbf{y})) \tag{2.27}$$

$$= \frac{1}{\mathcal{Z}_\theta}\exp\left(\theta^T\sum_{k=1}^{K}[\psi_k(x_k,y_k)+\phi_k(y_{k-1},y_k)]\right) \tag{2.28}$$

$$= \frac{1}{\mathcal{Z}_\theta}\prod_{k=1}^{K}\exp\left(\theta^T\psi_k(x_k,y_k)\right)\exp\left(\theta^T\phi_k(y_{k-1},y_k)\right) \tag{2.29}$$

$$= \prod_{k=1}^{K}\frac{1}{\mathcal{Z}_\theta^{\psi_k}(y_k)}\exp\left(\theta^T\psi_k(x_k,y_k)\right)\frac{1}{\mathcal{Z}_\theta^{\phi_k}(y_{k-1})}\exp\left(\theta^T\phi_k(y_{k-1},y_k)\right) \tag{2.30}$$

$$= \prod_{k=1}^{K}p_\theta(x_k|y_k)p_\theta(y_k|y_{k-1}) \tag{2.31}$$

où

$$\mathcal{Z}_\theta^{\psi_k}(y_k) = \sum_{x_k\in V}\exp\left(\theta^T\psi_k(x_k,y_k)\right) \tag{2.32}$$

$$\mathcal{Z}_\theta^{\phi_k}(y_{k-1}) = \sum_{y_k\in\mathcal{Y}(x_k)}\exp\left(\theta^T\phi_k(y_{k-1},y_k)\right) \tag{2.33}$$

et l'on retrouve le HMM bien connu, dont la factorisation obtenue peut être vue comme celle d'un modèle graphique orienté comme celui de la colonne centrale en

haut sur la figure 2.1. Le passage de l'équation 2.29 à l'équation 2.30 n'est pas trivial, mais résulte de la transformation d'un graphe non dirigé en un graphe dirigé en absence de "structures en V". On prouve ce résultat par induction en partant des feuilles et en normalisant localement à chaque étape (Wainwright et Jordan, 2008).

CRF Les calculs sont quasiment les mêmes en normalisant conditionnellement à $\mathbf{x} \in \mathcal{X}$

$$p_\theta(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathcal{Z}_\theta(\mathbf{x})} \exp(\theta^T \phi(\mathbf{x}, \mathbf{y})) \tag{2.34}$$

$$= \frac{1}{\mathcal{Z}_\theta(\mathbf{x})} \exp\left(\theta^T \sum_{k=1}^{K} [\psi_k(x_k, y_k) + \phi_k(y_{k-1}, y_k)]\right) \tag{2.35}$$

$$= \frac{1}{\mathcal{Z}_\theta(\mathbf{x})} \prod_{k=1}^{K} \exp\left(\theta^T \psi_k(x_k, y_k)\right) \exp\left(\theta^T \phi_k(y_{k-1}, y_k)\right) \tag{2.36}$$

Remarque 2.3.4. *Il est intéressant de remarquer que les modèles HMM et CRF,* pour un même jeu de traits*, sont exactement les mêmes. Ce qui diffère, c'est la manière dont on estime leurs paramètres. On considère souvent que les CRF sont plus riches que les HMM, ce qui est le cas dans la mesure où on peut concevoir un jeu de traits pour lequel le calcul est impossible pour un HMM*[7]*.*

2.4 D'autres critères d'apprentissage pour les modèles linéaires

Nous avons présenté les modèles exponentiels dans la section précédente, en nous limitant principalement au cas des modèles discriminants, qui définissent un modèle de probabilité conditionnel $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ sur les sorties sachant l'entrée. Mais comme nous l'avons discuté dans les sections 2.2.3 et 2.2.8, dans l'apprentissage discriminant, seule importe la fonction de décision $\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$ et d'autres méthodes sont possibles pour estimer le vecteur de paramètres.

Il existe plusieurs manières d'estimer les paramètres qui dépendent en particulier de l'objectif que l'on se fixe. Nous avons déjà rencontré l'estimation par le maximum de vraisemblance, qui est peut-être la manière la plus standard pour l'estimation de ces modèles. Remarquons que dans la littérature le terme de modèle log-linéaire ou de modèle exponentiel sous-entend le plus souvent que les

7. Entendre ici HMM étendu ou encore, comme illustré sur la figure 2.1 « modèle génératif général ».

paramètres sont estimés par maximum de vraisemblance conditionnelle (Smith et Eisner, 2005).

On peut interpréter l'approche du maximum de vraisemblance dans le cadre de la minimisation du risque empirique de la section 2.2.6 en choisissant comme fonction de perte (Smith, 2011, Section 3.3) :

$$perte_{gen}(\mathbf{x}, \mathbf{y}; f_{\boldsymbol{\theta}}) = -\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) \tag{2.37}$$

$$perte_{discr}(\mathbf{x}, \mathbf{y}; f_{\boldsymbol{\theta}}) = -\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) \tag{2.38}$$

Remarquons que cette formulation montre que de manière surprenante la fonction de perte ne dépend pas, du moins de manière explicite, de la fonction de prédiction.

Un des avantages à noter est que cette fonction objectif est convexe et donc relativement aisée à optimiser, aussi a-t-elle été utilisée dans de nombreuses tâches.

Cependant, d'autres fonctions de pertes sont possibles et seront particulièrement intéressantes lorsque nous chercherons à appliquer ces modèles pour la traduction automatique. Pour les tâches de TAL, et plus particulièrement lorsque nous nous intéressons à la traduction automatique, on peut identifier deux principaux inconvénients à utiliser la log-vraisemblance comme fonction objectif :

- La log-vraisemblance n'est pas forcément bien corrélée aux mesures d'évaluations de la tâche en question. Les problèmes de TAL impliquent généralement des tâches où les sorties prédites ne sont pas intégralement justes ou intégralement fausses : la log-vraisemblance ne prend pas en compte la mesure d'évaluation finale.
- Il est nécessaire d'être directement capable de calculer la probabilité de la référence, autrement dit d'être capable de calculer $\boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$. Pour la traduction automatique ceci n'est pas toujours possible et pose une réelle difficulté.

Nous présentons maintenant différentes fonctions de perte usuelles.

2.4.1 Le perceptron

L'algorithme du perceptron est peut-être l'une des méthodes les plus simples pour l'optimisation des paramètres, dont l'implémentation est triviale si l'on est capable de résoudre le problème d'inférence. Cela revient à considérer la fonction de perte suivante :

$$perte(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \boldsymbol{\theta}) = -\boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) + \max_{\mathbf{y} \in \mathcal{Y}(\tilde{\mathbf{x}})} \boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \mathbf{y}) \tag{2.39}$$

Cette fonction de perte est positive et vaut zéro si et seulement si le score de la référence $\tilde{\mathbf{y}}$ est meilleur que celui de tous les compétiteurs $\mathbf{y} \in \mathcal{Y}(\tilde{\mathbf{x}})$. À cause

de l'opérateur max, l'équation (2.39) n'est pas différentiable, mais reste cependant convexe et on peut considérer un sous-gradient. L'algorithme du perceptron tel que présenté ici a été étendu par Collins (2002) pour le cas structuré et utilisé depuis avec succès dans de nombreuses applications en TAL.

Dans l'équation (2.39), on peut remplacer l'opérateur max par l'opérateur $\log \sum \exp$, que l'on peut considérer comme une version lissée du maximum [8] et que nous appelons max-doux (*soft-max* en anglais). Dans ce cas, l'équation (2.39) devient

$$perte(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \boldsymbol{\theta}) = -\boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) + \log \sum_{\mathbf{y} \in \mathcal{Y}(\tilde{\mathbf{x}})} \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \mathbf{y})\right) \tag{2.40}$$

$$= -\log \frac{\exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})\right)}{\sum_{\mathbf{y} \in \mathcal{Y}(\tilde{\mathbf{x}})} \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \mathbf{y})\right)} \tag{2.41}$$

$$= -\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) \tag{2.42}$$

qui n'est autre que la perte de log-vraisemblance d'un modèle exponentiel. Cette correspondance entre le perceptron et les modèles exponentiels sera utile par la suite et nous y reviendrons dans le chapitre 3.

On peut remarquer une différence entre le max et le max-doux. Pour un exemple $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ donné, la log-vraisemblance ne serait maximisé que si l'intégralité de la masse de probabilité était attribué à la référence. Le modèle tend donc à augmenter au maximum le score de la référence. Dans le cas du max simple en revanche, il suffit que la référence ait un meilleur score que toutes les autres hypothèses, peu importe l'écart entre la référence et le meilleur des concurrents.

On comprend que cela peut poser un problème dans le cas où il y a plusieurs références acceptables. Dans le cas du perceptron, il est possible que toutes les références aient un bon score, à condition que la référence observée soit au moins légèrement meilleure. Le max-doux en revanche impose le score le plus faible possible à toutes les hypothèses différentes de la référence. Cela montre les limites de l'utilisation de la log-vraisemblance pour la traduction, et justifie de chercher à considérer de nouvelles fonctions de perte.

2.4.2 Intégration d'une fonction de coût lors de l'optimisation

L'algorithme du perceptron offre une méthode alternative d'optimisation mais ne fait pas de différence entre les différents compétiteurs. Une manière d'intégrer la

8. Au sens où cet opérateur est différentiable et si l'un des termes est très grand par rapport aux autres alors $\log \sum \exp$ vaut à peu près ce maximum.

fonction de coût (§ 2.2.7) est de considérer une fonction de perte dite « charnière » (*hinge-loss*) (Smith, 2011).

$$perte(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \boldsymbol{\theta}) = -\boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) + \max_{\mathbf{y} \in \mathcal{Y}(\tilde{\mathbf{x}})} \left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \mathbf{y}) + coût(\tilde{\mathbf{y}}, \mathbf{y}) \right) \tag{2.43}$$

Il ne s'agit alors plus de prendre la meilleure prédiction selon le modèle et de la comparer à la référence, mais de prendre la prédiction la plus « dangereuse », qui réalise un compromis entre un bon score selon le modèle et un coût élevé. Cette prédiction est parfois appelée l'hypothèse *peur* (*fear*) (Chiang, 2012).

En s'inspirant de l'analogie entre la log-vraisemblance et le perceptron, Gimpel et Smith (2010) ont proposé une version « douce » de l'équation (2.43), en remplaçant l'opérateur max par l'opérateur $\log \sum \exp$.

$$perte(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \boldsymbol{\theta}) = -\boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) + \log \sum_{\mathbf{y} \in \mathcal{Y}(\tilde{\mathbf{x}})} \exp \left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \mathbf{y}) + coût(\tilde{\mathbf{y}}, \mathbf{y}) \right) \tag{2.44}$$

Le problème de tous ces critères est qu'ils supposent que l'on soit en mesure de calculer $\boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$. Cela ne pose pas de problème particulier en classification binaire ou multi-classe, où le vecteur de caractéristiques s'exprime comme un produit tensoriel. Pour la plupart des tâches de TAL, ce calcul est également possible. Mais la spécificité de la traduction, où le vecteur de caractéristique n'est plus nécessairement défini pour tout couple d'entrée et sortie, impose de se tourner vers de nouveaux critères.

2.4.3 Fonctions de perte de douce rampe

On peut considérer une formulation un peu plus générale de la fonction de perte dite de « rampe » (*ramp-loss*). La forme générale est (Gimpel et Smith, 2012) :

$$\begin{aligned} perte(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \boldsymbol{\theta}) = \quad & - \bigoplus_{\mathbf{y}^+ \in \mathcal{Y}^+(\tilde{\mathbf{x}})}^{+} \boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \mathbf{y}^+) + \beta^+ gain^+(\tilde{\mathbf{y}}, \mathbf{y}^+) \\ & + \bigoplus_{\mathbf{y}^- \in \mathcal{Y}^-(\tilde{\mathbf{x}})}^{-} \boldsymbol{\theta}^T \boldsymbol{\phi}(\tilde{\mathbf{x}}, \mathbf{y}^-) + \beta^- coût^-(\tilde{\mathbf{y}}, \mathbf{y}^-) \end{aligned} \tag{2.45}$$

où $\bigoplus^+$, resp. $\bigoplus^-$, désignent soit l'opérateur *max* soit l'opérateur max-doux $\log \sum \exp$ [9] et $\beta^+, \beta^- \in \{0, 1\}$.

9. On pourrait considérer d'autres opérateurs ici.

En fonction du choix de *gain*, *coût*, β, $\bigoplus$, et $\mathcal{Y}$ on peut retrouver plusieurs pertes classiques (Gimpel et Smith, 2012) :

$$\begin{array}{lllll}
\text{log-MV :} & -\max_{\boldsymbol{\theta}} \boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\mathbf{x}},\tilde{\mathbf{y}}) & + & \log \sum_{\mathbf{y}\in\mathcal{Y}(\tilde{\mathbf{x}})} \exp\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\mathbf{x}},\mathbf{y})\right) \\
\text{perceptron :} & -\max_{\boldsymbol{\theta}} \boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\mathbf{x}},\tilde{\mathbf{y}}) & + & \max_{\mathbf{y}\in\mathcal{Y}(\tilde{\mathbf{x}})} \boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\mathbf{x}},\mathbf{y}) \\
\text{max-doux :} & -\max_{\boldsymbol{\theta}} \boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\mathbf{x}},\tilde{\mathbf{y}}) & + & \log \sum_{\mathbf{y}\in\mathcal{Y}(\tilde{\mathbf{x}})} \exp\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\mathbf{x}},\mathbf{y}) + \mathit{coût}(\tilde{\mathbf{y}},\mathbf{y})\right) \\
\text{max-marge :} & -\max_{\boldsymbol{\theta}} \boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\mathbf{x}},\tilde{\mathbf{y}}) & + & \max_{\mathbf{y}\in\mathcal{Y}(\tilde{\mathbf{x}})} \left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\mathbf{x}},\mathbf{y}) + \mathit{coût}(\tilde{\mathbf{y}},\mathbf{y})\right) \\
\text{risque-Jensen :} & -\max_{\boldsymbol{\theta}} \log \sum_{\mathbf{y}\in\mathcal{Y}(\tilde{\mathbf{x}})} \exp\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\mathbf{x}},\mathbf{y})\right) & + & \log \sum_{\mathbf{y}\in\mathcal{Y}(\tilde{\mathbf{x}})} \exp\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\mathbf{x}},\mathbf{y}) + \mathit{coût}(\tilde{\mathbf{y}},\mathbf{y})\right)
\end{array}$$

2.5 Optimisation

Dans cette partie, nous détaillons l'algorithme de propagation résiliente (Rprop) (Riedmiller et Braun, 1993) adapté pour pouvoir être utilisé avec une régularisation $\mathcal{L}_1$ (Lavergne *et al.*, 2010a). Comme l'extension de cet algorithme pour traiter la régularisation $\mathcal{L}_1$ n'a pas été préalablement décrite, nous en faisons ici une présentation rapide. L'idée est d'adapter la méthode de (Andrew et Gao, 2007) pour intégrer la régularisation $\mathcal{L}_1$. Rappelons que le problème général d'optimisation avec régularisation $\mathcal{L}_1$ et $\mathcal{L}_2$ est :

$$\arg\min_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta},\mathcal{D}) + \lambda_1\|\boldsymbol{\theta}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\theta}\|_2^2 \tag{2.46}$$

où $\boldsymbol{\theta} \in \mathbb{R}^K$, $\mathcal{D}$ est le corpus d'apprentissage et $\ell(\boldsymbol{\theta},\mathcal{D})$ est une fonction objectif dont on suppose que l'on sait calculer le gradient. Par exemple $\ell(\boldsymbol{\theta},\mathcal{D}) = \sum_{i=1}^{N} \mathit{perte}(\tilde{\mathbf{x}}^{(i)},\tilde{\mathbf{y}}^{(i)},\boldsymbol{\theta})$.

L'idée de l'algorithme Rprop est de considérer une descente de gradient dont le pas est ajusté indépendamment suivant chaque composante de $\boldsymbol{\theta}$. L'algorithme 1 montre son déroulement. Les lignes $1-4$ initialisent, pour chaque direction, le vecteur de paramètres θ_k, le gradient suivant cette direction g_k et le pas suivant cette direction η_k. Pour chaque itération de la descente de gradient (ligne 6), on considère chaque direction indépendamment (ligne 7) pour mettre à jour le gradient, le pas et la coordonnée du paramètre à apprendre. Les lignes $8-15$ permettent de calculer un pseudo-gradient suivant l'idée de Andrew et Gao (2007). Si θ_k est contenu dans l'intérieur d'un orthant, on peut prendre le gradient de la norme 1 (lignes $11-12$). Sinon, il faut choisir un pseudo-gradient (lignes $12-14$) étant

donné que la norme 1 n'est pas différentiable en 0. Le gradient est maintenu à 0, sauf si sa norme est supérieur à λ_1. On comprend alors pourquoi la régularisation $\mathcal{L}_1$ permet d'obtenir des paramètres dont de nombreuses composantes sont nulles : pour qu'une coordonnée ne soit pas nulle, il faut que sa contribution au gradient soit relativement importante, en l'occurrence plus grande que λ_1. Les lignes $16-23$ constituent la mise à jour proprement dite. Si le gradient a changé de signe (ligne 16), cela indique que l'on a dépassé le point d'optimum. On revient alors en arrière (ligne 17) et on divise la valeur du pas par deux (ligne 18). Sinon, on effectue la mise à jour (ligne 21) et, comme la direction prise semble bonne, on accélère la vitesse en augmentant le pas (ligne 22). Andrew et Gao (2007) proposent une étape de projection dans l'orthant d'origine après la mise à jour, mais dans nos expériences cette étape ne s'est pas avérée nécessaire. Remarquons que contrairement à la plupart des méthodes de descentes de gradient, seul le signe du gradient est utilisé (ligne 21), le pas étant uniquement déterminé par la séquence d'essais-erreurs.

2.6 Modèles à variables latentes

Lorsque l'on cherche à modéliser la probabilité d'un certain événement complexe, celle-ci peut parfois s'exprimer plus aisément à l'aide de variables explicatives cachées, que l'on appelle aussi latentes ou encore non observées. Par exemple, modéliser la répartition de la taille d'une population peut être plus simple en considérant les variables latentes `homme` et `femme`.

Dans le cas de la traduction automatique, modéliser directement le processus de traduction semble extrêmement compliqué. Il est donc utile d'introduire des variables latentes, dont on spécifie la forme, ce qui revient à introduire une forme de connaissance linguistique et ainsi à biaiser le modèle. En traduction automatique, ces variables latentes, que l'on appelle aussi dérivations, peuvent prendre différentes formes qui dépendent de l'approche choisie et proviennent de l'hypothèse d'un mécanisme sous-jacent au processus de traduction, comme par exemple la décomposition en segments ou le mouvement de certains mots.

Les variables latentes interviennent dans de nombreux problèmes de TAL. L'algorithme EM est souvent utilisé dans ce cas, mais nous allons ici plutôt considérer l'optimisation directe de la log-vraisemblance conditionnelle. En présence de variables latentes ($\mathbf{d}$), celle-ci s'exprime par :

Algorithme 1 : Adaptation de l'algorithme Rprop en intégrant la régularisation $\mathcal{L}_1$ et $\mathcal{L}_2$.

```
  /* Initialisation                                                    */
1 pour k ← 1 à K faire
2 |  θ_k^(0) ← 0.0;
3 |  g_k^(0) ← 0.0;
4 |  η_k^(0) ← 0.1;
5 fin
  /* Pour chaque itération                                             */
6 pour t ← 1 à T faire
  |  /* Pour chaque caractéristique                                   */
7 |  pour k ← 1 à K faire
  |  |  /* Calcul du gradient projeté sur un orthant                  */
8 |  |  g_k^(t) ← ∂f/∂θ_k(θ^(t−1), D) + λ2 θ_k^(t−1);
9 |  |  si λ1 > 0 alors
10|  |  |  si θ_k^(t−1) > 0 alors  g_k^(t) ← g_k^(t) + λ1 ;
11|  |  |  sinon si θ_k^(t−1) < 0 alors  g_k^(t) ← g_k^(t) − λ1 ;
12|  |  |  sinon si g_k^(t) > λ1 alors  g_k^(t) ← g_k^(t) − λ1 ;
13|  |  |  sinon si g_k^(t) < −λ1 alors  g_k^(t) ← g_k^(t) + λ1 ;
14|  |  |  sinon  g_k^(t) ← 0.0 ;
15|  |  fin
  |  |  /* Mise à jour                                                */
16|  |  si g_k^(t) × g_k^(t−1) < 0 alors
17|  |  |  θ_k^(t) ← θ_k^(t−1);
18|  |  |  η_k^(t) ← 0.5 η_k^(t−1);
19|  |  fin
20|  |  sinon si g_k^(t) × g_k^(t−1) > 0 alors
21|  |  |  θ_k^(t) ← θ_k^(t−1) + sgn(g_k^(t)) × η_k^(t) ;
22|  |  |  η_k^(t) ← 1.2 η_k^(t−1);
23|  |  fin
24|  fin
25 fin
```

$$\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \log \sum_{\mathbf{d} \in D(\mathbf{x})} p_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{d}|\mathbf{x}) \tag{2.47}$$

$$= \log \sum_{\mathbf{d} \in D(\mathbf{x})} \frac{1}{\mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x})} \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}, \mathbf{d})\right) \tag{2.48}$$

$$= \log \sum_{\mathbf{d} \in D(\mathbf{x})} \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}, \mathbf{d})\right) - \log \mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x}) \tag{2.49}$$

Savoir calculer le gradient de

$$\ell(\boldsymbol{\theta}, \mathcal{D}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}^{(i)}|\tilde{\mathbf{x}}^{(i)}) = \sum_{i=1}^{N} \log \sum_{\mathbf{d} \in D(\tilde{\mathbf{x}}^{(i)})} p_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}^{(i)}, \mathbf{d}|\tilde{\mathbf{x}}^{(i)}) \tag{2.50}$$

revient donc essentiellement à savoir calculer des espérances sur des ensembles de dérivations, représentés par des automates ou des forêts d'analyse.

L'optimum est ainsi atteint lorsque $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}, \mathcal{D}) = 0$, c'est-à-dire

$$\sum_{i=1}^{N} \mathbb{E}_{p(\mathbf{d}|\tilde{\mathbf{y}}^{(i)}, \tilde{\mathbf{x}}^{(i)})} \left[\boldsymbol{\phi}(\tilde{\mathbf{x}}^{(i)}, \tilde{\mathbf{y}}^{(i)}, \mathbf{d})\right] - \mathbb{E}_{p(\mathbf{y}, \mathbf{d}|\tilde{\mathbf{x}}^{(i)})} \left[\boldsymbol{\phi}(\tilde{\mathbf{x}}^{(i)}, \mathbf{y}, \mathbf{d})\right] \tag{2.51}$$

Nous allons faire usage des modèles à variables latentes dans deux contextes : au chapitre 3 pour modéliser l'ambiguïté dans une tâche d'apprentissage faiblement supervisé où l'information d'entrée n'est que partielle et au chapitre 5 pour la tâche de traduction automatique pour laquelle les variables latentes modélisent les dérivations sous-jacentes non observées.

2.7 Conclusions

Dans ce chapitre nous avons brièvement introduit deux tâches de traitement automatique des langues. L'analyse morpho-syntaxique sera au centre de nos intérêts dans la première partie de ce manuscrit : dans un contexte de langues peu dotées en ressources au chapitre 3 ; et comme exemple principal lors de l'étude de l'espace de recherche des modèles exponentiels au chapitre 4. La traduction automatique fera l'objet de la deuxième partie de notre travail, aux chapitres 5, 6 et 7.

Nous avons rappelé les principes de l'apprentissage automatique, avec une attention particulière pour l'apprentissage structuré, et de l'apprentissage génératif et discriminant. Nous avons décrit les modèles exponentiels, avec éventuellement

des variables latentes, qui nous serviront comme modèles de base pour l'apprentissage ambigu dans le chapitre 3, comme objet d'étude au chapitre 4 et qui sont au cœur des systèmes de traduction, comme nous le verrons au chapitre 5. Nous avons également discuté du choix de fonctions de pertes dans le cadre de la minimisation du risque empirique. Altun *et al.* (2003) s'intéressent, dans le cadre discriminant, à l'impact du choix de la fonction de perte et des méthodes d'optimisation. Leurs travaux semblent indiquer que le choix de la fonction de perte ou de la méthode d'optimisation n'a pas une influence très significative sur les performances. En revanche, le choix des caractéristiques du modèle influe considérablement sur les résultats obtenus. Cette conclusion est partagée par Klein et Manning (2002), qui suggèrent que le principal intérêt de l'apprentissage discriminant est justement de permettre l'introduction de caractéristiques qui ne peuvent pas être utilisées simplement dans les modèles génératifs. La tendance en TAL depuis une dizaine d'années, vers l'utilisation de caractéristiques plus expressives ou parallèlement vers l'introduction d'un très grand nombre de caractéristiques simples, semble confirmer ces observations.

Signalons enfin un domaine de l'apprentissage automatique dont l'importance au sein la communauté de TAL ne cesse de croître récemment et dont nous n'avons pas parlé dans ce travail : celui des réseaux de neurones et de l'apprentissage profond. Les modèles de réseaux de neurones permettent d'apprendre automatiquement des représentations compactes et distribuées, au lieu d'avoir à identifier, plus ou moins manuellement, les caractéristiques adaptées à une tâche en particulier.

Chapitre 3

Apprentissage ambigu : application au transfert cross-lingue

Sommaire

Dans ce chapitre nous allons nous intéresser à un cadre moins favorable que celui donné par l'apprentissage supervisé standard, que nous avons décrit au chapitre 2. Nous supposons maintenant que l'on dispose, lors de l'apprentissage, uniquement d'observations partielles sur la sortie de référence, au lieu de connaître celle-ci complètement comme dans le cadre supervisé. Ce cadre d'apprentissage « ambigu », où l'on ne connaît plus avec précision la référence, peut englober plusieurs applications. Par exemple, en traduction automatique, on ne connaît généralement pas la bonne segmentation de référence, mais on dispose pourtant de l'ensemble ambigu de toutes les segmentations acceptables (celles qui mènent à la phrase cible de référence) comme nous le discuterons au chapitre 5. Dans ce chapitre, nous nous intéressons plus particulièrement à une autre tâche, celle de l'analyse morpho-syntaxique pour des langues peu dotées. En utilisant un cadre de transfert cross-lingue, dans lequel les annotations pour la langue cible d'intérêt sont projetées à partir d'une langue source plus riche en annotations, et en combinant ces informations avec des dictionnaires extraits automatiquement, il est possible de produire des annotations — en général incomplètes et ambiguës — pour la langue cible. Dans ce chapitre, nous étudions l'extension possible de deux méthodes d'apprentissage structuré pour prendre en compte ce cadre ambigu et étudions leur comportement. Nous nous intéressons également plus particulièrement à divers aspects de la tâche en question, évaluons l'importance des différentes ressources en jeu et effectuons différentes analyses qui nous permettent de mettre en lumière les difficultés de l'évaluation de ce cadre. Ce chapitre, fruit d'une collaboration avec Guillaume Wisniewski et mes directeurs de thèse, a été publié en partie dans (Wisniewski *et al.*, 2014b) pour les toutes premières expériences et dans (Wisniewski *et al.*, 2014a) pour l'évaluation plus complète sur dix langues de familles différentes. Ce chapitre est une extension plus détaillée de ces travaux, reprise dans (Pécheux *et al.*, 2016b), où nous proposons plusieurs expériences de contrôle et contrastons diverses configurations qui nous permettent de mieux cerner les raisons et conditions dans lesquelles l'apprentissage est possible et efficace. La question étudiée dans ce chapitre concerne principalement celle de l'espace de supervision, et la possibilité que celui-ci soit ambigu. Le problème de l'espace de recherche, pour l'étude proposée ici, sera traité dans le chapitre 4.

3.1 Introduction

Lorsque l'on dispose de peu ou d'aucunes données annotées pour une tâche particulière, comme c'est par exemple le cas pour de nombreuses langues faiblement dotées en ressources, il n'est pas possible d'appliquer telles quelles les méthodes d'apprentissage supervisées habituellement utilisées en TAL. Il est alors nécessaire de chercher d'autres formes d'annotations, le plus souvent incomplètes et/ou

de moins bonne qualité, par exemple en utilisant des dictionnaires extraits automatiquement ou en transférant des annotations à partir d'une autre langue. Ces méthodes de transfert cross-lingue permettent partiellement de pallier l'absence de corpus annotés. Le transfert d'étiquettes morpho-syntaxiques depuis une langue riche en ressources, complété et corrigé par un dictionnaire associant à chaque mot un ensemble d'étiquettes autorisées, ne fournit cependant qu'une information de supervision incomplète. Dans ce chapitre, nous nous intéressons à la possibilité d'apprendre de manière effective à partir d'annotations partielles, ce que nous formalisons dans le cadre de l'*apprentissage ambigu* (Bordes *et al.*, 2010; Cour *et al.*, 2011). Nous étudions en détail les extensions possibles de deux modèles supervisés : d'une part un modèle CRF, déjà proposé dans certains travaux antérieurs ; et introduisons d'autre part une extension d'un modèle à base d'historique. Nous utilisons une tâche d'analyse morpho-syntaxique comme cas d'étude, mais de nombreux résultats nous semblent pouvoir s'interpréter de manière plus générale. Des expériences menées sur dix langues, appartenant à différentes familles linguistiques, montrent que nous parvenons à des résultats sensiblement meilleurs que ceux de l'état de l'art, parfois de manière assez marquante. De bonnes performances peuvent ainsi être atteintes, même en présence d'ambiguïté, à condition cependant de disposer à la fois de ressources monolingues et bilingues. Nous étudions en détail les conditions pour lesquelles un apprentissage ambigu est effectivement possible et observons que les deux méthodes étudiées présentent chacune des avantages et des inconvénients, suivant les cas. Bien au-delà du choix du critère d'apprentissage, de nombreux paramètres peuvent se révéler critiques pour obtenir de bonnes performances. Enfin, nous étudions le cadre d'évaluation d'une telle tâche et montrons que des différences de conventions d'annotations rendent délicate l'évaluation précise du cadre de transfert cross-lingue.

Le point de départ de cette étude est issu de (Täckström *et al.*, 2013a). Ces auteurs s'intéressent à une tâche d'analyse morpho-syntaxique pour des langues cibles peu dotées, pour lesquelles deux types de ressources sont disponibles : d'une part un dictionnaire (Wiktionnaire) permettant de connaître, pour un mot, l'ensemble de ses catégories morpho-syntaxiques possibles ; d'autre part, un corpus parallèle aligné mot-à-mot et dont la partie source a été étiquetée automatiquement. En combinant ces deux ressources, comme nous le verrons dans la section 3.3.5, les auteurs montrent qu'il est possible d'apprendre un analyseur morpho-syntaxique de bonne qualité, même lorsque l'on ne dispose pas directement de données cibles annotées.

Dans ce chapitre, nous reproduisons le cadre faiblement supervisé de Täckström *et al.* (2013b) afin de mieux comprendre leurs résultats et présentons une analyse détaillée du comportement de leur CRF partiellement observé et d'un nouveau modèle à base d'historique plus simple, mais aussi efficace. À condition d'utili-

ser au mieux les différentes ressources pendant l'apprentissage et l'inférence, les deux méthodes dépassent les meilleurs résultats publiés jusqu'ici pour dix langues de différentes familles, dans certains cas de manière importante. De nombreuses expériences de contrôle et une analyse des erreurs nous permettent de conclure que cette amélioration dépend de nombreux facteurs et doit être analysée avec précaution.

Les principales contributions de ce chapitre sont :

- La formalisation du problème d'apprentissage avec transfert cross-lingue d'annotations comme un problème d'apprentissage ambigu, ce qui nous a conduit à proposer une nouvelle approche ;
- Une comparaison empirique de deux méthodes d'apprentissage et la définition précise des conditions dans lesquelles celles-ci obtiennent de bons résultats ;
- De nombreuses expériences de contrôle permettant de mieux comprendre à quel point il est possible d'apprendre en présence d'ambiguïté, le rôle et l'importance de cette ambiguïté ainsi que de mettre en lumière les compromis entre les différentes sources possibles ;
- Une analyse des erreurs de nos systèmes qui nous a permis de soulever le problème des conventions d'annotation et la difficulté de l'évaluation d'un tel cadre.

L'un des points importants que nous soulevons est que les méthodes présentées ici nécessitent la combinaison de données parallèles et monolingues pour être effectivement utilisables, ce qui rend finalement peu certaine leur applicabilité dans un véritable contexte de langues peu dotées.

Le reste de ce chapitre est structuré de la manière suivante : nous présentons d'abord plus en détail le cadre du transfert cross-lingue ainsi que les différents travaux qui ont été consacrés à ce problème (§ 3.2). Nous présentons ensuite en détail les différentes ressources qui peuvent être utilisées pour inférer automatiquement des annotations partielles (§ 3.3). Nous nous intéressons ensuite au cadre de l'apprentissage ambigu et exposons les extensions de deux méthodes supervisées pour leur permettre de prendre en compte l'ambiguïté des annotations obtenues (§ 3.4). Nous présentons enfin toute une série d'expériences à la fois pratiques et de contrôle dans la section 3.5, ainsi qu'une analyse d'erreur. La section 3.6 nous permet de résumer brièvement nos principales conclusions.

3.2 Apprentissage cross-lingue et langues peu dotées : état des lieux

Au cours de ces deux dernières décennies, les méthodes d'apprentissage supervisé se sont imposées pour de nombreuses tâches de TAL. Les méthodes d'étiquetage automatique utilisées par exemple pour l'analyse en parties du discours atteignent aujourd'hui un niveau de performances proche de celui d'un annotateur humain, du moins lorsqu'elles sont entraînées sur des corpus annotés suffisamment grands dans le domaine d'intérêt Manning (2011). Leur succès dépend donc de manière cruciale de la disponibilité d'une quantité suffisante de données annotées, ce qui est loin d'être toujours le cas. D'autres méthodes pouvant se passer de données annotées ou pouvant du moins se contenter d'annotations partielles ont donc été recherchées.

Les données de supervision peuvent prendre différentes formes. Les travaux de Garrette et Baldridge (2013) et Duong *et al.* (2014a) ont montré que des annotations manuelles, même lorsque qu'elles ne sont disponibles qu'en faible quantité, restent très efficaces lorsqu'il s'agit de guider l'induction de bons analyseurs.

L'annotation manuelle d'un corpus reste cependant un processus complexe, fastidieux et onéreux qui nécessite une solide expertise linguistique (Abeillé *et al.*, 2003), même si les outils aujourd'hui disponibles peuvent aider à accélérer très significativement cette démarche (Garrette et Baldridge, 2013). Il n'existe donc actuellement de corpus annotés avec des informations morpho-syntaxiques que pour un nombre de langues et de domaines réduits. Différentes approches ont été proposées dans la littérature pour réduire cet effort d'annotation (voire pour s'en passer complètement) afin de développer des analyseurs morpho-syntaxiques pour des langues et des domaines pour lesquels ces ressources n'existent pas. Ainsi, de nombreuses approches ont été développées récemment pour combler l'absence de données annotées pour une langue cible peu dotée.

Le besoin de pouvoir disposer d'outils de TAL pour un très grand nombre de langues, sans avoir besoin de développer de nouvelles ressources adaptées à chaque tâche est l'un des problèmes les plus importants de ces dernières années et qui a donné lieu à de nombreux développements. Diverses techniques ont été développées pour un grand nombre de tâches et de langues, et notre objectif n'est pas ici de fournir une étude exhaustive, mais plutôt de se concentrer sur les études les plus emblématiques dans le cadre de l'analyse morpho-syntaxique.

Les techniques ne nécessitant aucune annotation, c'est-à-dire complètement non-supervisées, pour l'analyse morpho-sytaxique datent au moins des études de Mérialdo (1994), qui utilise l'algorithme EM, en ayant éventuellement recours à un faible nombre de données annotées pour restreindre localement le nombre d'étiquettes possibles pour un certain mot-type. Malgré de nombreuses tentatives,

vingt ans plus tard, les performances des méthodes non-supervisées restent très en deçà des résultats que l'on peut attendre lorsque l'on dispose d'annotations de référence (Christodoulopoulos *et al.*, 2010). Ceci a motivé de nombreuses recherches pour obtenir des données, en recourant à diverses méthodes pour construire automatiquement des annotations, éventuellement imparfaites et incomplètes.

Un premier type de solution consiste à estimer des classes de mots automatiquement à partir de corpus non annotés, en regroupant les unités qui possèdent un même comportement distributionnel ; ces classes doivent ensuite être projetées sur les catégories morpho-syntaxiques traditionnelles pour pouvoir être interprétées. Une grande variété de méthodes ont été proposées dans la littérature pour réaliser cette tâche, depuis (Brown *et al.*, 1992) jusqu'aux travaux plus récents de Banko et Moore (2004); Toutanova et Johnson (2007). Malgré des progrès constants, leurs performances restent encore trop faibles pour permettre leur utilisation dans des applications de TAL (Christodoulopoulos *et al.*, 2010). Cette approche peut être largement améliorée dès lors que l'on dispose d'une poignée de données annotées en plus des données non étiquetées (*apprentissage semi-supervisé*) : les annotations serviront, par exemple, à initialiser et/ou à désambiguïser les catégories apprises automatiquement.

Dans les approches précédentes, pour projeter les mots sur une liste de catégories prédéfinies, on utilise des dictionnaires qui contraignent la liste des étiquettes possibles de chaque mot. Mérialdo (1994) remarque que ces dictionnaires, qui permettent de réaliser une désambiguïsation partielle, peuvent se révéler extrêmement utiles pour guider les méthodes d'apprentissage non supervisé — par exemple dans un cadre de modèle à données latentes. Ceci est largement confirmé par (Li *et al.*, 2012b), où les auteurs construisent avec succès des analyseurs morpho-syntaxiques pour huit langues en utilisant un HMM non-supervisé à caractéristiques riches (Berg-Kirkpatrick *et al.*, 2010). De tels dictionnaires peuvent aujourd'hui être obtenus automatiquement à un coût relativement bas (Li *et al.*, 2012b), par exemple à partir des données de projets tels que Wiktionnaire[1] — une source d'information que nous exploiterons également abondamment et dont nous reparlerons en détail à la section 3.3.4. Leur travail inclut également une étude empirique des performances lorsque davantage d'entrées de Wiktionnaire sont prises en compte.

Lorsque l'on ne dispose d'aucune annotation manuelle pour une langue cible, il est également possible d'obtenir relativement facilement des annotations à partir d'autres langues similaires si l'on dispose de données annotées dans une autre langue. C'est l'esprit des méthodes par *transfert direct*. Il est possible soit de directement transférer les annotations, par exemple à partir de liens d'alignement, et ensuite d'apprendre un modèle pour la langue cible sur ces données, soit d'ap-

1. http://www.wiktionary.org/

prendre un modèle sur la langue source et de transférer les *paramètres* de ce modèle pour la langue cible. Les caractéristiques utilisées sont alors le plus souvent non-lexicales, afin que les phrases en langue source soient les plus semblables à celles en langue cible (Zeman et Resnik, 2008). Ces approches peuvent être complétées par des étapes supplémentaires qui visent à calibrer plus précisément les modèles en utilisant des informations de la langue cible visée de manière non-supervisée, comme par exemple dans (Cohen *et al.*, 2011), où de nombreuses langues sources sont d'ailleurs simultanément considérées.

Le transfert cross-lingue offre ainsi une autre manière, complémentaire, de contourner l'absence ou la rareté de données annotées. Le principe du transfert cross-lingue est d'exploiter des corpus de textes parallèles, qui peuvent aujourd'hui être collectés automatiquement en grande quantité (Resnik et Smith, 2003) et d'utiliser ceux-ci pour *transférer* les sorties des outils d'analyse appliqués à une langue *source* riche en données annotées vers une langue *cible* moins bien dotée. Ainsi, en exploitant les alignements automatiques au niveau des mots, il est possible de projeter les étiquettes morpho-syntaxiques des phrases sources vers les phrases cibles (Yarowsky *et al.*, 2001). Cette approche a été initiée par Yarowsky *et al.* (2001), qui étudie le transfert de diverses annotations syntaxiques. Ce travail a inspiré de nombreux travaux ultérieurs, également influencés par l'importance considérable que ce domaine revêt de nos jours et par le besoin toujours plus croissant d'entraîner des outils de TAL malgré l'absence de données annotées. On peut citer quelques études récentes, par exemple (Hwa *et al.*, 2005; Ganchev *et al.*, 2009; Zhao *et al.*, 2009; Durrett *et al.*, 2012) pour des dépendance syntaxiques, (Padó et Lapata, 2009; Kozhevnikov et Titov, 2013; van der Plas *et al.*, 2014) pour l'étiquetage en rôle sémantique ou encore (Kim *et al.*, 2012; Wang et Manning, 2014a) pour la reconnaissance d'entités nommées.

Cependant ce processus de transfert des annotations de la langue source vers la langue cible à travers les liens d'alignements ne permet d'obtenir, en général, que des annotations partielles et relativement bruitées pour la langue cible. En faisant l'hypothèse que les annotations peuvent effectivement être projetées d'une langue à une autre, des techniques nouvelles doivent être trouvées pour prendre en compte le caractère incomplet et/ou incertain des annotations collectées, en particulier en utilisant des alignements dont la qualité n'est pas toujours garantie.

En particulier, l'approche de (Täckström *et al.*, 2013b), que nous reprenons ici, vise à filtrer les erreurs conséquentes à de mauvais alignements en contraignant les informations projetées à l'aide d'informations extraites à partir de dictionnaires monolingues. La méthode de transfert des étiquettes entre langue source et langue cible que nous utilisons, ainsi que l'extraction et l'utilisation des dictionnaires, sont détaillées dans la partie 3.3.

Das et Petrov (2011) développent une autre méthode pour réduire l'effet du

bruit dans les alignements. Les corpus parallèles sont ici utilisés pour construire des listes d'étiquettes transférées (uniquement à partir de liens considérés comme sûrs) pour des trigrammes de mots cibles. Une propagation des étiquettes est ensuite utilisée pour généraliser cette information à d'autres trigrammes cibles. Une méthode non-supervisée à la Berg-Kirkpatrick *et al.* (2010) est ensuite utilisée sur les données cibles ainsi étiquetées, en utilisant les étiquettes ambiguës comme caractéristiques.

Les travaux récents de (Ganchev et Das, 2013a) peuvent également être vus comme une autre tentative d'exprimer l'incertitude au niveau des étiquettes. Les étiquettes transférées sont utilisées en tant que contraintes flexibles comme régularisation d'une méthode non supervisée, suivant le cadre de la régularisation a *posteriori* (*posterior regularization*) (Ganchev *et al.*, 2010).

Wang et Manning (2014b) montrent qu'il est également possible et préférable de transférer les probabilités calculées par les outils d'analyse en langue source plutôt que de projeter uniquement les étiquettes prédites.

Une dernière possibilité est d'utiliser des corpus parallèles artificiels, que l'on peut obtenir par exemple en traduisant automatiquement des documents de la langue cible vers une langue source riche en annotations (Tiedemann, 2014). Cette approche permet ainsi d'éliminer les problèmes d'erreurs dans les alignements, au prix, cependant, d'erreurs possibles lors du processus de traduction. De plus, cette approche suppose également que l'on dispose de systèmes de traduction automatique pour la langue et le domaine d'intérêt.

Cependant, il n'existe à l'heure actuelle aucune analyse comparative qui permette clairement de comprendre les raisons du succès de telles méthodes, ni de mesurer l'importance de meilleures techniques d'apprentissage, par rapport à l'obtention d'annotations partielles plus adaptées ou de meilleure qualité.

Les deux approches proposées par Täckström *et al.* (2013b), permettant d'apprendre à partir de ces deux sources de données (les étiquettes projetées et les dictionnaires), reposent sur des modèles de séquences (HMM et CRF) et sur une généralisation *ad hoc* de leur critère d'apprentissage, afin d'intégrer les différentes sources d'information. Dans ce travail, nous proposons également de reformuler le problème du transfert cross-lingue dans le cadre de l'*apprentissage ambigu* (Bordes *et al.*, 2010; Cour *et al.*, 2011) dont l'objectif est d'estimer un classifieur lorsque le système ne peut accéder, lors de la phase d'apprentissage, qu'à un ensemble d'étiquettes possibles dont une seule est juste et non à l'étiquette de référence. À partir des résultats théoriques développés dans (Bordes *et al.*, 2010), nous introduisons une méthode d'apprentissage capable d'apprendre un étiqueteur morpho-syntaxique dans un contexte faiblement supervisé. Ce modèle d'apprentissage est décrit dans la partie 3.4 et son évaluation pour dix langues est présentée dans la partie 3.5.

3.3 Création partielle de corpus d'apprentissage par transfert d'étiquettes

L'objectif de ce travail est de développer des étiqueteurs morpho-syntaxiques en s'appuyant sur le transfert d'annotations entre phrases parallèles, afin de pouvoir complètement se dispenser, lors de l'apprentissage, de données étiquetées manuellement. Le transfert d'annotations nécessite de définir une correspondance entre étiquettes des langues source et cible, correspondance qui est obtenue dans ce travail en utilisant un ensemble universel d'étiquettes morpho-syntaxiques simples, décrit dans la sous-section 3.3.1. La qualité de ces étiquettes transférées étant sensible aux erreurs d'alignement et d'étiquetage de la phrase source (Yarowsky *et al.*, 2001), plusieurs travaux, en particulier (Täckström *et al.*, 2013b), ont proposé de combiner ces étiquettes avec des informations issues de données monolingues afin d'éviter au maximum les séquences d'étiquettes invalides. À l'instar de ces travaux, nous utilisons deux sources complémentaires d'information pour déterminer les étiquettes des mots de la langue cible par transfert cross-lingue : un dictionnaire associant à un mot-type donné l'ensemble de ses étiquettes possibles (§ 3.3.4) et les alignements entre une phrase annotée et sa traduction (§ 3.3.3). Ces informations sont ensuite combinées pour étiqueter automatiquement un corpus d'apprentissage (§ 3.3.5). La figure 3.1 résume cette approche.

3.3.1 Un ensemble universel d'étiquettes morpho-syntaxiques

La possibilité de transférer l'information morpho-syntaxique d'une langue à une autre suppose que cette information puisse être décrite de la même manière dans les deux langues. Même si cette hypothèse forte est hautement controversée (Evans et Levinson, 2009; Broschart, 2009), Petrov *et al.* (2012a) définissent 12 étiquettes [2], morpho-syntaxiques à gros grain choisies en raison de leur « universalité » (les catégories identifiées sont relativement stables d'une langue à l'autre) et de leur utilité dans une chaine de traitement de TAL. Ces étiquettes universelles sont les suivantes : NOUN (noms), VERB (verbes), ADJ (adjectifs), ADV (adverbes), PRON (pronoms), DET (déterminants et articles), ADP (prépositions et postpositions), NUM (numéraux), CONJ (conjonctions), PRT (particules), « . » (symboles de ponctuations) et X (pour tout ce qui échappe aux autres catégories, comme par exemple les abréviations ou les mots étrangers). Ces catégories sont uniquement décrites par des exemples et par leur association à des corpus exis-

2. Cet ensemble d'étiquettes universelles à été étendu récemment, au sein d'un projet plus général (*Universal Dependencies*, `http://universaldependencies.org`) visant à proposer des corpus arborées universels pour un très grand nombre de langues (Nivre *et al.*, 2015).

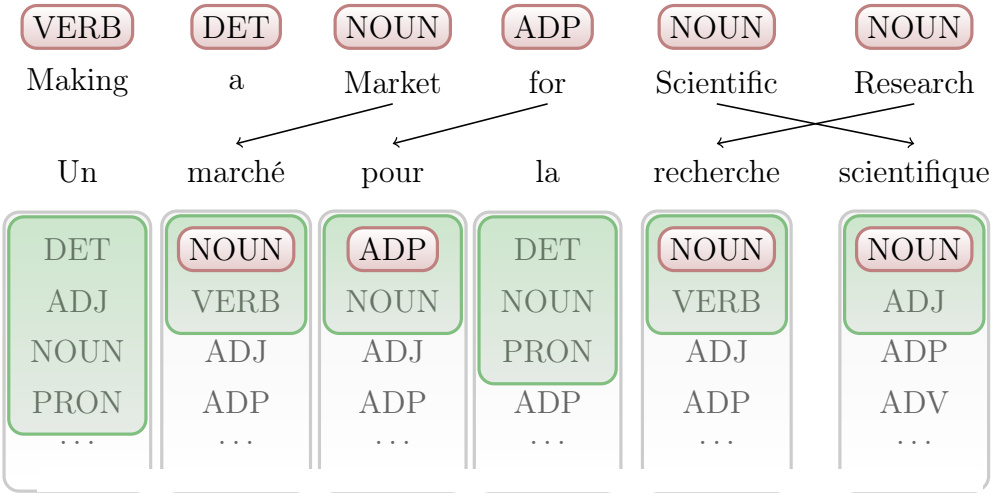


FIGURE 3.1 – Exemple de transfert d'étiquettes d'une phrase source (haut) en anglais vers une phrase cible (bas) en français, extrait du corpus d'apprentissage. Pour chaque mot cible, les étiquettes autorisées par les contraintes de types sont représentées dans le cadre en vert. Les étiquettes morpho-syntaxiques de la phrase source sont transférées vers la phrase cible uniquement lorsque celles-ci sont « compatibles » avec les contraintes de types.

tants et n'ont pas vraiment fait l'objet d'une caractérisation formelle. Par la suite, nous supposerons toujours que toutes les étiquettes morpho-syntaxiques ont été transformées en étiquettes universelles.

3.3.2 Qualité et ambiguïté des contraintes obtenues

Dans un contexte de transfert cross-lingue, plusieurs sources d'information peuvent intervenir pour inférer automatiquement des étiquettes morpho-syntaxiques. L'étiquetage ainsi constitué peut cependant être à la fois incomplet (les étiquettes ne sont pas inférées pour tous les mots d'une phrase), bruité (l'étiquetage ainsi inféré peut contenir des erreurs) ou encore ambigu (plusieurs étiquettes peuvent être inférées pour un même mot, sans que l'on sache *a priori* laquelle est correcte). Les différentes sources d'information utilisées pour construire l'étiquetage de référence peuvent donc, pour un corpus donné, être caractérisées de la manière suivante :

— leur *ambiguïté*, que nous décrivons par deux indicateurs : le pourcentage de mots-occurrences ambigus (c'est-à-dire le pourcentage d'occurrences comprenant strictement plus d'une étiquette) et l'ambiguïté moyenne, c'est-à-dire le nombre moyen d'étiquettes par mot-occurrence ;

— leur *précision*, c'est-à-dire le pourcentage de mots-occurrences dont l'une des étiquettes inférées est correcte ;[3]

— leur *couverture*, qui correspond au nombre de mots-occurrences pour lesquelles on possède une information, c'est-à-dire qui sont au moins partiellement désambiguïsés.

Remarquons qu'il est toujours possible d'augmenter la précision aux dépens de l'ambiguïté : il suffit d'associer aux mots l'ensemble de toutes les étiquettes possibles, l'ambiguïté et la précision sont alors maximales.

Par la suite, nous allons considérer deux sources d'information que l'on peut regrouper en deux familles principales : les *contraintes d'occurrences* et les *contraintes de types*, qui sont décrites ci-dessous.

3.3.3 Transfert cross-lingue d'étiquettes

La première source d'information, appelée *contrainte d'occurrence* (en anglais *token constraint*), utilise les liens d'alignements, lorsqu'ils existent, pour projeter l'étiquette d'un mot-occurrence source sur un mot-occurrence cible. Les liens d'alignements peuvent être automatiquement calculés à l'aide de méthodes non supervisées, en utilisant les modèles IBM implantés dans MGIZA (Och et Ney, 2003; Gao et Vogel, 2008) (voir § 5.3). En général, on ne dispose pas de corpus parallèles annotés, mais il est possible d'apprendre en premier lieu un analyseur pour la langue riche (en utilisant un autre corpus annoté) et en utilisant ce dernier pour inférer automatiquement les étiquettes des phrases sources du bitexte. Il est d'usage (Das et Petrov, 2011; Täckström *et al.*, 2013b) de ne considérer que des alignements un-à-un, ce qui permet de transférer directement, pour chaque mot-occurrence aligné, l'étiquette du mot source vers le mot cible associé. La figure 3.1 montre un exemple d'alignement entre une phrase source et une phrase cible. L'alignement manquant entre « a » et « Un » ne permet pas de transférer l'étiquette DET. En suivant (Täckström *et al.*, 2013b), nous appelons cette source d'information des *contraintes d'occurrences*, puisque cela induit une contrainte sur les étiquettes possibles à une position d'une phrase donnée. L'un des avantages de cette méthode est de permettre un étiquetage avec une ambiguïté minimale pour les mots couverts (en effet, ces mots sont alors étiquetés par une seule étiquette). La précision de cette méthode dépend en revanche de plusieurs facteurs : de la qualité de l'annotation de la langue source, des erreurs d'alignement et/ou de divergences linguistiques entre les deux langues considérées.

3. Cet indicateur ne peut être calculé que pour des données dont on connaît un étiquetage de référence.

On peut illustrer ce dernier cas par un exemple (voir également (Dorr, 1994) pour d'autres exemples de divergences systématiques) :

Anglais : She swam across the lake
Français : Elle a traversé le lac à la nage

où un lien d'alignement pourrait associer le verbe anglais « swam » avec le nom français « nage » et la préposition anglaise « across » avec le verbe français « traversé ». Enfin, la couverture des contraintes d'occurrences est liée à la similarité entre les langues source et cible. Elle est en fait égale à la proportion de mots-occurrences cibles alignés. Le tableau 3.2 montre que cette couverture est de l'ordre de 70% et varie peu suivant les langues [4].

D'autres difficultés, qui dépendent par exemple de schémas structurels différents suivant les langues, peuvent également apparaître. Dans les corpus arborés que nous utilisons dans nos expériences (voir § 3.5.1), en anglais, des quantificateurs comme 'few' ou 'little' sont généralement utilisés en conjonction avec un déterminant ('a few years', 'a little parable', ...) et étiquetés comme ADJ ; la construction correspondante en espagnol fait disparaître l'article ('*mucho tiempo*', '*pocos años*', ...) et les quantificateurs sont ainsi étiquetés par DET. La capture de différences aussi subtiles ne semble pas simple sans connaissances *a priori* et sans l'utilisation de caractéristiques spécialement dédiées.

3.3.4 Utilisation de dictionnaires

La deuxième source d'information utilisée pour prédire les informations morpho-syntaxiques est appelée *contrainte de type* et se fonde sur un dictionnaire qui associe à chaque mot-type l'ensemble des étiquettes autorisées pour ce mot. La figure 3.1 donne un exemple d'étiquettes autorisées pour une phrase en français. Pour tous les mots de cet exemple, le dictionnaire de types permet de limiter les étiquettes possibles à deux, trois ou quatre alternatives. Comme expliqué dans la partie 3.3.5, ces contraintes permettent de réduire les étiquettes possibles pour chaque mot et de filtrer les annotations transférées en suivant les liens d'alignements.

Les dictionnaires peuvent avoir diverses origines. Les corpus bitextes peuvent constituer une première source possible. À partir des alignements mot à mot, il est possible de collecter pour chaque mot-type l'ensemble des étiquettes qui sont transférées sur l'une de ses occurrences. Afin d'éviter d'entacher ces contraintes d'erreurs dues aux alignements ou aux événements rares, il est utile de les filtrer. En pratique, pour un mot-type donné, nous incluons une étiquette parmi les contraintes de types lorsque celle-ci apparaît dans au moins 10% des alignements.

4. L'anglais est ici toujours la langue source, mais il pourrait être avantageux dans certains cas d'avoir une langue source plus proche de la langue cible considérée.

FIGURE 3.2 – Exemple de page de WIKTIONNAIRE pour le mot « monte » qui peut être à la fois un nom ou une forme de verbe. Les deux flèches rouges indiquent les en-têtes qui contiennent les informations de catégorie morpho-syntaxique.

Nous appelons ces contraintes les *contraintes bitextes*. Le tableau 3.1 montre que le dictionnaire ainsi constitué se caractérise par une très grande couverture et une ambiguïté relativement faible. En revanche, il semble peu fiable d'utiliser ce dictionnaire pour filtrer les liens d'alignements lors du transfert cross-lingue, puisque le dictionnaire est construit à partir des mêmes données [5].

Une autre source possible pour les contraintes de types est d'extraire automatiquement des informations du Web et de profiter des outils collaboratifs. On peut par exemple dériver des dictionnaires de types à partir de WIKTIONNAIRE, un dictionnaire multi-lingue, gratuit et à grande échelle, dont les entrées contiennent entre autres des informations sur les parties du discours et sur la prononciation. Dans ce travail, nous utilisons des dictionnaires extraits automatiquement de WIKTIONNAIRE en utilisant les méthodes et les heuristiques introduites par Li *et al.* (2012b).

WIKTIONNAIRE [6], un projet dérivé de `Wikipedia`, est un projet collaboratif,

5. Mais ce dictionnaire permet toutefois de filtrer les projections à travers les alignements les plus rares ou les plus bruités, puisque nous avons utilisé une heuristique de filtrage, qui peut s'apparenter d'ailleurs à une forme de cross-validation.

6. `http://wiktionary.org`

multi-lingue, offrant un dictionnaire gratuit dans de très nombreuses langues. En octobre 2014, WIKTIONNAIRE affiche plus de 21 millions d'entrées pour 171 langues, 33 d'entre elles possédant plus de 100,000 entrées [7]. Ce nombre important d'entrées est principalement dû au fait que WIKTIONNAIRE définit des entrées à la fois pour les lemmes (comme dans un dictionnaire standard) mais également pour les variantes morphologiques obtenues à partir d'inflexions comme les déclinaisons ou les conjugaisons. La figure 3.2 montre qu'une entrée de WIKTIONNAIRE peut contenir bon nombre d'informations, depuis la définition, les parties du discours ou la prononciation jusqu'à l'étymologie, les anagrammes, hyponymes, hyperonymes, ainsi que des traductions.

Les pages de WIKTIONNAIRE contiennent également une autre source très utile d'information : la liste des variantes morphologiques pour un grand nombre de formes. Cette information est généralement donnée sous la forme d'un patron d'inflexion qui utilise la possibilité de concaténation des chaînes de caractères de WIKTIONNAIRE pour engendrer automatiquement les différentes formes en fonction de règles. Par exemple, le patron `{{es.v.conj.hacer|h}}` permet de dériver toutes formes du verbe espagnol *hacer* (faire), soit une bonne soixantaine. En changeant le paramètre par `desh`, on obtient les formes pour le verbe *deshacer* (défaire).

Prendre en compte toutes les variantes morphologiques des mots permet d'augmenter de manière très significative la couverture du dictionnaire extrait, en particulier pour les langues à fort caractère inflexionnel. Par exemple, pour le finnois, le processus d'extraction simple qui ne considère que les formes ayant leur propre page WIKTIONNAIRE permet d'extraire 51, 418 contraintes de types, tandis que lorsque l'on utilise WIKTIONNAIRE pour engendrer les variantes morphologiques, on obtient 455, 568 contraintes, soit une augmentation d'un facteur presque 10.

Le tableau 3.1 montre que ce dictionnaire possède une moins bonne couverture sur le corpus d'apprentissage que les contraintes qui sont extraites de ces corpus, mais permet cependant d'améliorer la couverture sur le corpus de test. On s'attend également à ce que ces contraintes contiennent moins d'erreurs ; la précision est d'ailleurs meilleure sur le corpus de test. Li *et al.* (2012b) étudient également en détail la couverture et l'exactitude des étiquettes morpho-syntaxiques extraites de WIKTIONNAIRE.

Lorsque plusieurs dictionnaires sont disponibles comme c'est le cas ici, il peut également être pertinent de les combiner, afin de tirer profit de leur complémentarité. Dans ce travail, nous considérons deux stratégies différentes : pour chaque mot-type qui apparaît à la fois dans les contraintes de WIKTIONNAIRE et les contraintes bitextes, nous pouvons prendre soit l'union, soit l'intersection des contraintes. Si un mot-type apparaît uniquement dans l'un ou l'autre des

7. On peut retrouver ces statistiques et bien d'autres sur WIKTIONNAIRE à l'adresse https://meta.wikimedia.org/wiki/Wiktionary

		ar	cs	de	el	es	fi	fr	id	it	sv
amb.	wiki	11.4	3.6	2.4	3.2	2.0	4.0	2.4	3.0	2.6	2.6
	bitexte	1.6	1.3	1.5	1.4	1.3	1.4	1.4	1.4	1.5	1.4
	wiki ∪ bitexte	1.6	1.5	1.8	1.7	1.7	1.7	2.0	1.6	1.8	2.0
	wiki ∩ bitexte	1.6	1.1	1.2	1.2	1.1	1.2	1.1	1.1	1.2	1.2
amb.	wiki +occ.	5.9	1.4	1.4	1.6	1.3	1.7	1.8	1.4	1.4	1.4
	bitexte +occ.	1.5	1.1	1.2	1.2	1.1	1.2	1.2	1.1	1.2	1.2
	wiki ∪ bitexte +occ.	1.5	1.2	1.4	1.3	1.3	1.3	1.5	1.2	1.3	1.3
	wiki ∩ bitexte +occ.	1.5	1.1	1.1	1.1	1.1	1.2	1.1	1.1	1.1	1.1
>1	wiki	94.6	34.0	34.0	39.8	39.5	38.9	49.4	36.5	39.5	46.2
	bitexte	31.9	26.9	29.9	36.7	22.7	23.9	35.0	25.0	37.0	32.6
	wiki ∪ bitexte	32.4	35.7	47.3	49.3	50.2	40.9	61.8	45.2	55.1	52.0
	wiki ∩ bitexte	31.3	9.1	6.6	16.3	8.9	11.2	11.3	6.5	14.8	16.0
>1	wiki +occ.	44.8	9.8	15.5	17.6	17.9	12.1	25.5	8.3	15.4	16.6
	bitexte +occ.	17.5	8.6	12.7	14.8	5.5	9.5	16.2	8.1	14.0	11.2
	wiki ∪ bitexte +occ.	17.8	10.8	20.0	20.0	17.1	14.1	27.8	11.0	20.1	16.1
	wiki ∩ bitexte +occ.	17.2	3.5	3.2	8.4	4.8	5.2	6.1	2.2	7.0	6.8
couv.	wiki	5.5	77.3	91.0	82.5	94.5	74.7	92.6	84.2	88.9	92.1
	bitexte	97.5	99.8	99.3	99.8	99.9	98.9	99.9	99.6	99.9	99.8
	wiki ∪ bitexte	97.5	99.8	99.4	99.9	99.9	99.1	99.9	99.7	99.9	99.8
	wiki ∩ bitexte	97.5	99.8	99.4	99.9	99.9	99.1	99.9	99.7	99.9	99.8

(a) Corpus d'apprentissage

		ar	cs	de	el	es	fi	fr	id	it	sv
amb.	wiki	10.4	3.7	2.4	3.2	2.2	4.3	2.4	3.3	2.8	2.5
	bitexte	3.1	2.5	2.1	2.0	2.0	3.3	1.6	2.7	2.1	2.1
	wiki ∪ bitexte	3.1	2.4	2.2	1.9	2.1	3.1	2.1	2.5	2.2	2.4
	wiki ∩ bitexte	3.1	2.0	1.6	1.5	1.5	2.6	1.3	2.1	1.6	1.7
>1	wiki	85.4	35.2	32.1	39.1	38.9	42.9	47.2	31.3	39.6	44.9
	bitexte	37.8	35.7	32.7	39.1	28.9	39.2	35.4	37.7	39.5	36.9
	wiki ∪ bitexte	38.3	40.7	46.4	47.6	49.6	47.9	59.4	42.0	53.5	55.6
	wiki ∩ bitexte	37.0	15.1	9.0	17.2	12.4	21.4	12.0	13.9	17.4	19.2
couv.	wiki	14.7	76.3	90.1	82.0	93.0	72.0	92.3	80.5	87.4	92.3
	bitexte	83.4	88.8	92.9	94.8	92.8	81.3	98.0	87.2	93.8	93.2
	wiki ∪ bitexte	83.6	91.7	95.0	96.9	96.7	85.9	98.7	90.5	95.8	95.3
	wiki ∩ bitexte	83.6	91.7	95.0	96.9	96.7	85.9	98.7	90.5	95.8	95.3
préc.	wiki	93.0	97.7	94.9	94.6	94.9	96.2	94.5	90.3	95.0	96.9
	bitexte	79.6	89.4	92.5	91.5	91.3	89.1	92.8	91.8	93.8	93.8
	wiki ∪ bitexte	79.7	97.6	97.6	97.5	97.9	96.0	97.6	94.2	98.2	98.6
	wiki ∩ bitexte	78.7	94.2	92.7	93.0	93.5	93.3	91.7	89.4	93.3	94.0

(b) Corpus de test

Tableau 3.1 – Statistiques, au niveau des occurrences, sur les corpus d'apprentissage (a) et sur les corpus de test (b) décrits à la section 3.5.1 pour différentes combinaisons possibles de contraintes de types en appliquant éventuellement également les contraintes d'occurrences (+occ) : ambiguïté moyenne (amb.) ; pourcentage d'occurrences ambiguës (> 1) ; couverture (couv.) ; et précision (préc.).

	ar	cs	de	el	es	fi	fr	id	it	sv
% ali.	53.0	77.8	66.7	69.3	74.0	73.1	64.7	81.6	72.2	79.9
% accord	96.5	83.7	83.9	78.3	82.9	83.2	83.8	80.5	82.1	86.0
% utile	27.6	8.6	6.0	14.6	6.6	9.9	9.6	6.4	13.2	13.4

Tableau 3.2 – Statistiques, au niveau des occurrences, pour le corpus d'apprentissage en utilisant l'intersection des contraintes de types ($wiki \cap bitexte$) ainsi que les contraintes d'occurrences : pourcentage de mots-occurrences cibles alignés (ali.) (*i.e.* couverture des contraintes d'occurrences) ; pourcentage des contraintes d'occurrences compatibles avec les contraintes de types (accord) ; et pourcentage de contraintes d'occurrences « informatives » (utile), correspondant aux occurrences compatibles pour lesquelles (i) une étiquette est effectivement transférée et (ii) les contraintes de types seules n'auraient pas suffi pas à désambiguïser entièrement l'occurrence.

dictionnaires, on utilise alors les contraintes de ce dictionnaire. Täckström *et al.* (2013b) considèrent également le cas de l'union, mais pas celui de l'intersection. En considérant également l'un ou l'autre de ces dictionnaires seuls, on obtient quatre possibilités. Chaque condition traduit un compromis différent entre l'ambiguïté, la couverture et la précision, ce qui va nous permettre de mieux comprendre l'importance de chaque source d'information. Le tableau 3.1 présente différentes caractéristiques des combinaisons possibles de ces contraintes. Comme l'on peut s'y attendre, prendre l'intersection permet de réduire de manière importante l'ambiguïté, tant sur le corpus d'apprentissage que sur celui de test, mais au prix d'une précision moins élevée (l'étiquette correcte peut en effet être enlevée). Inversement, l'union permet d'augmenter la précision en test, en contrepartie d'une ambiguïté plus élevée [8].

3.3.5 Prise en compte des deux sources d'information

Si les deux sortes de contraintes ont été utilisées dans de nombreux travaux, Täckström *et al.* (2013b) sont les premiers à proposer de les combiner pour profiter de leur complémentarité. Nous reproduisons ici leur cadre, qui consiste à supposer que les contraintes de types sont plus sûres que les contraintes d'occurrences, ces

8. Il peut sembler paradoxal que la précision de l'union soit moins bonne que celle des contraintes de Wiktionnaire seules, mais il ne faut pas oublier que la couverture change également lorsque l'on combine les sources de contraintes. En particulier, lorsque la couverture de Wiktionnaire est faible (par exemple pour l'arabe), les erreurs des contraintes bitextes dominent le terme de précision qui diminue donc d'autant.

Algorithme 2 : Règles utilisées pour transférer les étiquettes à partir d'une phrase source.

Entrées : mot w, d dictionnaire décrivant les contraintes de types et un alignement entre les phrases source et cible
Sorties : l'ensemble des étiquettes possibles pour le mot w

```
1 occurrence ← {étiquette du mot avec lequel w est aligné} type ← d[w]
2 si type ∩ occurrence ≠ ∅ alors
3 |  retourner occurrence;
4 sinon
5 |  retourner type;
6 fin
```

dernières étant particulièrement sensibles aux erreurs d'alignement. Par défaut, on associe donc à chaque mot-type les étiquettes autorisées par les contraintes de types. Si aucune contrainte de type ne s'applique, alors on utilise l'ensemble des 12 étiquettes universelles. De plus, lorsque pour une occurrence donnée une étiquette provenant d'un mot source aligné devrait être projetée, celle-ci n'est effectivement prise en compte que si elle est compatible avec les contraintes de types. Cette procédure est illustrée sur la figure 3.1 qui donne un exemple de transfert et de filtrage des étiquettes d'une phrase source vers une phrase cible. Les deux sources d'information introduites précédemment sont fusionnées en utilisant les règles décrites par l'algorithme 2, qui s'inspire de la méthode de Täckström *et al.* (2013b). Remarquons que dans les rares cas où un mot-type n'est jamais aligné dans le corpus d'entraînement, nous utilisons le jeu complet d'étiquettes. Le tableau 3.1 montre que, si l'on utilise les contraintes bitextes, c'est le cas pour moins de 1% des mots-occurrences (sauf pour l'arabe pour lequel le taux est de 2.5%).

Les statistiques rassemblées dans le tableau 3.1 indiquent l'importance des contraintes d'occurrences. Pour toutes les langues, les contraintes d'occurrences permettent de réduire à la fois l'ambiguïté moyenne et le nombre de positions ambiguës, qui est réduit de plus de moitié quelle que soit la configuration des contraintes de types. Dans la configuration la plus favorable, l'ambiguïté moyenne pour un mot est à peine supérieure à 1 (environ 1.1 pour 8 sur les 10 langues), et plus de 92% des mots-occurrences sont complètement désambiguïsés (à l'exception de l'arabe pour lequel les statistiques sont nettement moins favorables). Une dernière observation positive est que le bruit est relativement limité (de l'ordre de 10% ou moins dans la plupart des cas, hormis celui de l'arabe, encore une fois).

Finalement, le tableau 3.2 permet de remarquer que les contraintes de types permettent bien de filtrer les erreurs dans les contraintes d'occurrences (de l'ordre

de 20% des cas).

Remarquons que dans cette approche, le dictionnaire joue un rôle central : d'une part, en validant les étiquettes projetées au travers des liens d'alignement pour créer la référence ; d'autre part, en restreignant l'espace de recherche de l'analyseur morpho-syntaxique. Lors de l'apprentissage et du décodage, la liste des étiquettes possibles pour chaque mot peut alors être réduite à un ensemble d'alternatives (les étiquettes autorisées par le dictionnaire) bien plus restreint que l'ensemble des étiquettes définies dans le schéma d'annotation.

Täckström *et al.* (2013b) décrivent de manière détaillée l'impact de ces deux types de contraintes et montrent que chacune d'elles apporte des informations complémentaires.

Après transfert et filtrage des étiquettes, les mots cibles sont donc associés à un ensemble d'étiquettes (en rouge, figure 3.1) et non à une unique étiquette de référence. Un mot-occurrence cible peut cependant être associé à une unique étiquette, dans le cas du transfert d'une étiquette ou dans le cas où la contrainte de type est réduite à une étiquette. Le tableau 3.1 montre que c'est le cas pour environ 80% des mots-occurrences. Dans la section suivante, nous expliquons comment il est possible d'entraîner un analyseur morpho-syntaxique n'utilisant que cette *information ambiguë* comme supervision.

3.4 Modèles de séquences pour l'apprentissage faiblement supervisé

Les contraintes que nous avons décrites dans la section précédente permettent d'associer à chaque mot un sous-ensemble d'étiquettes, parmi lesquelles une seule, au mieux, est correcte. Si cette situation est plus avantageuse que le cadre non-supervisé, du fait d'une information au moins partielle à chaque position, elle est clairement moins favorable que le cadre standard supervisé où l'on connaît exactement l'étiquette correcte. De plus, le fait que l'information de supervision soit ambiguë et incomplète ne permet plus d'utiliser telles quelles les méthodes d'apprentissage usuelles. La figure 3.3 montre un exemple d'instance d'apprentissage dans ce cadre.

Dans cette section, nous étudions brièvement comment ce problème a été envisagé par la communauté en apprentissage statistique (§ 3.4.1) avant de décrire deux extensions possibles des méthodes supervisées au cadre ambigu : la première (§ 3.4.2) part d'un modèle de champs aléatoires markoviens (CRF) et reproduit le modèle de Täckström *et al.* (2013b) ; la deuxième (§ 3.4.3), que nous avons introduite dans (Wisniewski *et al.*, 2014b,a), étend un modèle à base d'historique.

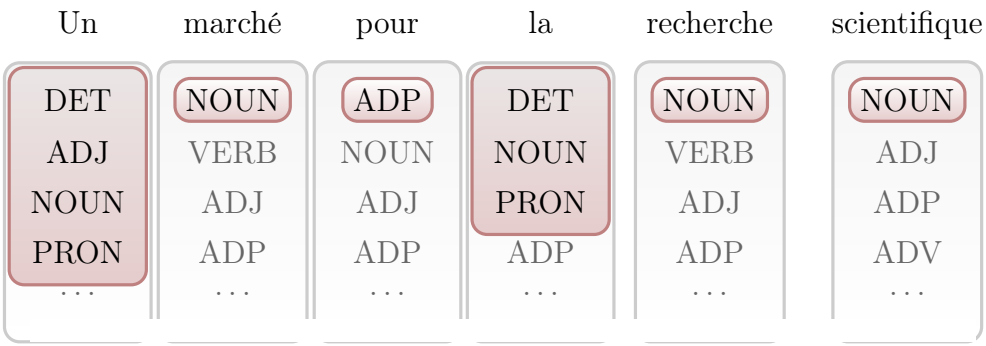


FIGURE 3.3 – Exemple d'instance d'apprentissage ambigu, obtenu après combinaison des contraintes de types et d'occurrences. Notons que pour certaines positions, l'étiquetage de référence est ambigu. Cette instance d'apprentissage est celle qui résulterait de la combinaison du transfert cross-lingue et des dictionnaires illustrés par la figure 3.1

3.4.1 Apprentissage ambigu

En apprentissage statistique, l'apprentissage ambigu désigne en général un cadre pour lequel chaque exemple d'entrée $\mathbf{x}$ est associé à plusieurs sorties possibles $\widetilde{\mathcal{Y}}$, parmi lesquelles une et une seule est correcte. Autrement dit, on suppose que l'observation sur étiquette correcte $\tilde{\mathbf{y}}$ a été corrompue par l'ajout d'étiquettes fallacieuses $\widetilde{\mathcal{Y}}\backslash\{\tilde{\mathbf{y}}\}$. La figure 3.3 présente une instance d'apprentissage ambigu obtenue après transfert cross-lingue dans l'exemple de la figure 3.1.

Ce cadre a cependant reçu bien d'autres noms dans la littérature, ce qui démontre plutôt la richesse de la variété d'approches qui ont été proposées pour le résoudre. Il a été décrit en premier lieu par (Jin et Ghahramani, 2002) comme un problème d'*étiquetage multiple*, étant donné que chaque exemple est associé à plusieurs étiquettes. (Tsuboi *et al.*, 2008; Cour *et al.*, 2011; Täckström *et al.*, 2013b) remarquent que par rapport au cadre supervisé, les annotations ne sont que partiellement observées et proposent de voir ce cadre comme un problème d'apprentissage *partiellement observé*. Finalement, (Bordes *et al.*, 2010; Cour *et al.*, 2011; Li *et al.*, 2012c) décrivent ce problème sous l'angle de l'*apprentissage ambigu*, mettant en avant le fait que chaque étiquette de référence est observée en compagnie d'autres compétiteurs fallacieux. Par la suite, nous utiliserons le terme *apprentissage ambigu*, étant donné que la méthode d'apprentissage proposée s'inspire directement des résultats de (Bordes *et al.*, 2010).

De nombreux scénarios peuvent se formaliser sous l'angle de l'apprentissage ambigu. Un problème similaire à celui considéré dans ce chapitre est l'étiquetage morpho-syntaxique lorsque les données d'annotation ont été obtenues à partir de

différents annotateurs, et peuvent donc différer à certains endroits (Jin et Ghahramani, 2002; Tsuboi *et al.*, 2008; Dredze *et al.*, 2009). Ce cadre comprend également la segmentation en mots (Tsuboi *et al.*, 2008), l'apprentissage de correspondances sémantiques (Bordes *et al.*, 2010) ou l'identification de visages dans des images ou des collections de vidéos (Cour *et al.*, 2011).

Nous commençons par formaliser le cas de l'apprentissage multi-classe, puis nous aborderons dans un second temps l'extension de ces concepts au cas structuré qui nous intéresse. Soit $\mathcal{U}$ l'ensemble de toutes les étiquettes possibles (ici les 12 étiquettes universelles). Pour une instance d'entrée x, notons par $\mathcal{Y}$ l'ensemble des étiquettes autorisées et par $\widetilde{\mathcal{Y}}$ les étiquettes de référence, où $\widetilde{\mathcal{Y}} \subseteq \mathcal{Y} \subseteq \mathcal{U}$. L'ensemble $\mathcal{Y}$ correspond en général à l'ensemble de toutes les étiquettes possibles $\mathcal{U}$, mais, comme nous le discuterons plus tard, en particulier dans le chapitre suivant, les contraintes de types peuvent être utilisées pour réduire l'ensemble des étiquettes possibles pour certains exemples x. Dans le cadre supervisé standard, $\widetilde{\mathcal{Y}}$ ne contient qu'un seul élément : l'étiquette de référence. Dans le cadre de l'apprentissage ambigu cependant, $\widetilde{\mathcal{Y}}$ est un ensemble qui peut contenir plusieurs éléments : l'étiquette de référence, ainsi qu'un certain nombre d'étiquettes fallacieuses (pour une occurrence donnée).

Étant donné que l'on ne dispose ici d'aucune information sur l'identité de l'étiquette de référence parmi les étiquettes de $\widetilde{\mathcal{Y}}$, la plupart, sinon toutes les méthodes d'apprentissage ambigu reviennent d'une manière ou d'une autre à s'assurer que l'on donne aux étiquettes de $\widetilde{\mathcal{Y}}$ un score au moins aussi bon que celles dans $\mathcal{Y} \setminus \widetilde{\mathcal{Y}}$. Étant donné que l'étiquette correcte est inconnue, une approche raisonnable est d'augmenter le score des étiquettes de $\widetilde{\mathcal{Y}}$, afin de rendre la prédiction de l'étiquette correcte plus probable. Les méthodes d'apprentissage diffèrent principalement par la manière dont cela est réalisé.

Une première tentative naïve, décrite dans Jin et Ghahramani (2002); Cour *et al.* (2011) revient à considérer que chaque étiquette de référence possible $\tilde{y} \in \widetilde{\mathcal{Y}}$ est correcte et conduit à construire un nouvel ensemble d'instances d'apprentissage dans lequel pour chaque observation $(\mathbf{x}, \widetilde{\mathcal{Y}})$ du corpus initial, un exemple $(\mathbf{x}, \tilde{y})$ est ajouté pour chaque $\tilde{y}$ dans $\widetilde{\mathcal{Y}}$. Cette méthode revient à concaténer toutes les paires instance/étiquette possibles. Les résultats expérimentaux montrent cependant que cette approche naïve ne s'accompagne pas de très bons résultats, probablement en raison de la quantité de bruit introduite par l'ajout de nombreuses instances d'apprentissage incorrectes.

Étant donné qu'une seule étiquette est effectivement correcte, une meilleure idée serait de permettre au modèle de trouver de lui-même un étiquetage de référence à partir duquel apprendre ses paramètres. Ceci peut être réalisé par une approche « à points de vue multiples » (Li *et al.*, 2012c) ou en utilisant l'algorithme EM comme dans (Jin et Ghahramani, 2002). Pour ce dernier cas, à partir de pa-

ramètres initiaux, l'algorithme calcule la probabilité *a posteriori* d'un étiquetage des exemples d'apprentissage, utilisée pour réestimer les paramètres du modèle, le tout étant itéré plusieurs fois. Dans ce modèle à variables latentes, on définit en général une forme paramétrique pour $p_\theta(y|\mathbf{x})$ (de paramètres θ) et on maximise la vraisemblance conditionnelle suivante :

$$p_\theta(\widetilde{\mathcal{Y}}|x) = \sum_{\tilde{y}\in\widetilde{\mathcal{Y}}} p_\theta(\tilde{y}|x) \tag{3.1}$$

Afin d'expliquer au mieux $\widetilde{\mathcal{Y}}$, il semble possible de s'attendre à ce que la valeur optimale des paramètres implique que l'étiquette correcte $\tilde{y}$ ait une probabilité plus importante que les autre étiquettes. Ce cadre peut s'étendre relativement simplement au cas de l'apprentissage structuré (Tsuboi *et al.*, 2008; Dredze *et al.*, 2009; Täckström *et al.*, 2013b) comme nous le verrons dans la section 3.4.2. Ce modèle surpasse la méthode naïve précédente dans les expériences de Jin et Ghahramani (2002) ; même si (Cour *et al.*, 2011) obtiennent des résultats plus nuancés.

Bordes *et al.* (2010) reformulent l'apprentissage ambigu comme un problème de classement. Dans un cadre entièrement supervisé, on cherche à s'assurer que l'étiquette de référence est mieux classée que toutes les alternatives. Par analogie, Bordes *et al.* (2010) définissent une *fonction de perte de classement ambiguë*, qui vise à s'assurer que tous les exemples positifs (ceux de $\widetilde{\mathcal{Y}}$) sont mieux classés que les exemples négatifs. Sous quelques hypothèses peu contraignantes (§ 3.4.3), il est possible de montrer que minimiser cette fonction de perte ambiguë revient à minimiser la fonction de perte 0/1 recherchée.

Un dernier travail similaire est celui de Cour *et al.* (2011), qui montre que certaines classes de fonctions de perte peuvent être étendues au cas ambigu. En utilisant une relaxation convexe de ces fonctions de perte, les auteurs obtiennent des résultats théoriques qui permettent de borner la fonction de perte du cas supervisé. Les auteurs utilisent dans leurs expériences une adaptation d'une fonction de perte à large marge, ce qui fonctionne bien. Notre travail s'inspire directement de Cour *et al.* (2011) mais nous adaptons plutôt une méthode de perceptron.

Dans la suite de cette section, nous expliquons comment il est possible de modifier deux algorithmes d'étiquetage standard — les champs markoviens aléatoires et un modèle à base d'historique — de façon à pouvoir les appliquer au cas d'un étiquetage de référence ambigu. Les notations que nous avons utilisées jusqu'ici peuvent être étendues au cas structuré de la manière suivante : soit $\mathbf{x} = \left(x_1, \ldots, x_{|\mathbf{x}|},\right)$ la séquence d'observation et $\mathcal{Y} = \left(\mathcal{Y}_1, \ldots, \mathcal{Y}_{|\mathbf{x}|}\right)$ et $\widetilde{\mathcal{Y}} = \left(\widetilde{\mathcal{Y}}_1, \ldots, \widetilde{\mathcal{Y}}_{|\mathbf{x}|}\right)$ les séquences (d'ensemble) d'étiquettes respectivement autorisées et de référence, où pour chaque position i, $\widetilde{\mathcal{Y}}_i \subseteq \mathcal{Y}_i \subseteq \mathcal{U}$.

3.4.2 Champs markoviens conditionnels partiellement observés

Les champs markoviens conditionnels (CRF) (Lafferty *et al.*, 2001; Sutton et McCallum, 2007; Tellier et Tommasi, 2011), que nous avons décrits à la section 2.3.4 ont été largement utilisés pour modéliser des problèmes d'étiquetage de séquences, atteignant des performances parmi les meilleurs résultats de la littérature pour de nombreuses tâches, en particulier l'analyse morpho-syntaxique. Rappelons que les CRF modélisent la probabilité d'une séquence d'étiquettes $\mathbf{y} = \left(y_1, \ldots, y_{|\mathbf{x}|}\right) \in \mathcal{Y}$ conditionnée par rapport à la séquence observée $\mathbf{x}$ en définissant un modèle exponentiel normalisé

$$p_\theta(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\theta(\mathbf{x})} \exp \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \tag{3.2}$$

où θ est le vecteur de paramètres, $\phi(\mathbf{x}, \mathbf{y})$ le vecteur de caractéristiques et

$$Z_\theta(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \exp \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \tag{3.3}$$

est la fonction de partition.

Dans le cadre standard de l'apprentissage complètement supervisé, le principe de maximum de vraisemblance conduit à choisir les paramètres qui maximisent la vraisemblance conditionnelle, compte tenu des données (entièrement observées) $\mathcal{D}_{\text{supervised}} = (\mathbf{x}^{(k)}, \tilde{\mathbf{y}}^{(k)})_{k=1}^K$. Dans ce chapitre, afin d'éviter les confusions, ce modèle CRF entièrement supervisé sera appelé CRF-S.

Dans le cas de l'apprentissage ambigu, il est possible d'élargir la définition de la vraisemblance pour que celle-ci comprenne le cas de données partiellement observées $\mathcal{D}_{\text{ambiguous}} = (\mathbf{x}^{(k)}, \widetilde{\mathcal{Y}}^{(k)})_{k=1}^K$:

$$\mathcal{L}_\theta = \sum_{k=1}^{K} \log p_\theta \left(\widetilde{\mathcal{Y}}^{(k)} | \mathbf{x}^{(k)} \right) \tag{3.4}$$

où

$$p_\theta(\widetilde{\mathcal{Y}}|\mathbf{x}) = \sum_{\tilde{\mathbf{y}} \in \widetilde{\mathcal{Y}}} p_\theta(\tilde{\mathbf{y}}|\mathbf{x}) \tag{3.5}$$

est la généralisation de l'équation (3.1) au cas structuré. Ce modèle faiblement supervisé sera appelé CRF-A.

L'optimisation de la fonction de perte (3.5) est un problème d'optimisation non-convexe, par opposition au cas supervisé où la vraisemblance conditionnelle est une fonction convexe des paramètres. Il est cependant possible d'utiliser une méthode

de descente de gradient pour atteindre un minimum qui ne sera éventuellement que local [9].

Intuitivement, le processus d'apprentissage vise à transférer la masse de probabilité de toutes les étiquettes possibles $\mathcal{Y}$ vers les étiquettes identifiées comme possibles références $\widetilde{\mathcal{Y}}$, sans imposer de contraintes particulières sur la manière dont la masse de probabilité se redistribue au sein des séquences de $\widetilde{\mathcal{Y}}$. On peut donc raisonnablement espérer que la méthode d'apprentissage sera à même d'identifier certains motifs dans les données d'apprentissage qui permettront à la séquence correcte de « s'approprier » la plus grande part de la masse de probabilité par rapport aux séquences erronées.

Comme discuté par (Täckström *et al.*, 2013b), l'espace de recherche $\mathcal{Y}$ (qui définit les étiquettes lors du calcul de la fonction de partition à l'équation (3.3)) n'est pas nécessairement l'ensemble complet de toutes les séquences d'étiquettes envisageables $\mathcal{U}^{|\mathbf{x}|}$. Il est possible d'utiliser les contraintes de types pour filtrer l'espace de recherche, tant à l'apprentissage que lors du décodage. Dans ce chapitre nous nous limitons à l'utilisation des contraintes de types pour restreindre l'espace de recherche lors du décodage, ce qui est d'ailleurs une pratique courante (Ratnaparkhi, 1996; Moore, 2014). Nous étudierons, dans le chapitre 4, l'utilisation de ces contraintes lors de l'apprentissage. Comme nous le verrons, ce choix n'est pas anodin et permet de très larges gains de performance par rapport à des configurations alternatives, largement moins avantageuses.

3.4.3 Un modèle à base d'historique

Les modèles à base d'historique (Black *et al.*, 1992) sont une autre méthode populaire pour prédire des séquences d'étiquettes. Nous proposons ici d'étendre un modèle à base d'historique (Black *et al.*, 1992; Collins, 2003; Tsuruoka *et al.*, 2011) avec une méthode d'apprentissage proche de LaSO (Daumé et Marcu, 2005). Les modèles à base d'historique réduisent la prédiction d'une structure à une séquence de problèmes de classification multi-classe. La prédiction d'une structure complexe (ici, une séquence d'étiquettes morpho-syntaxiques) est ainsi modélisée comme une suite de problèmes de décision : pour chaque position de la séquence, un classifieur multi-classe est utilisé pour prendre une décision, en utilisant des caractéristiques qui décrivent à la fois la structure d'entrée et l'historique des décisions passées (c'est-à-dire la séquence partiellement annotée).

Notons $\mathbf{x} = (x_i)_{i=1}^{n}$ la séquence d'observations et $\mathcal{Y}$ l'ensemble des étiquettes possibles (dans notre cas les 12 étiquettes universelles). L'inférence consiste à pré-

9. En pratique, nous n'avons pas observé de différence importante en variant les conditions initiales.

dire les étiquettes les unes après les autres en utilisant, ici, un modèle linéaire :

$$y_i^* = \arg\max_{y \in \mathcal{Y}} \boldsymbol{\theta}^T \phi(\mathbf{x}, i, y, h_i) \tag{3.6}$$

où y_i^* est l'étiquette prédite pour la i-ème observation, $\boldsymbol{\theta}$ le vecteur de poids, $h_i = y_1^*, ..., y_{i-1}^*$ l'historique décrivant les décisions passées à l'étape i et ϕ un vecteur de traits représentant de manière jointe l'observation, l'étiquette candidate et l'historique. Ainsi, l'inférence peut être vue comme une recherche gloutonne dans l'espace de tous les étiquetages possibles $\mathcal{Y}$ de la séquence d'entrée. Ce type de modèles, qui sacrifie un optimum global au bénéfice d'une plus grande flexibilité des traits [10], a été utilisé avec succès dans de nombreuses applications de TAL (Kazama et Torisawa, 2007; Ratinov et Roth, 2009; Tsuruoka *et al.*, 2011).

Algorithme 3 : Algorithme d'apprentissage. Dans le cas ambigu, $\tilde{\mathcal{Y}}_i$ est l'ensemble des étiquettes autorisées ; dans le cas supervisé, cet ensemble est réduit à l'étiquette de référence. Le nombre T d'itérations effectuées est un hyperparamètre de l'algorithme.

$\boldsymbol{\theta}_0 \leftarrow 0$;
pour $t \in [\![1, T]\!]$ **faire**
  tirer au hasard un exemple $\mathbf{x}, \tilde{\mathbf{y}}$;
  $h \leftarrow$ liste vide ; `// Initialise un historique vide`
  pour $i \in [\![1, |\mathbf{x}|]\!]$ **faire**
    $y_i^* = \arg\max_{y \in \mathcal{Y}_i} \boldsymbol{\theta}^T \phi(\mathbf{x}, i, y, h)$;
    si $y_i^* \notin \tilde{\mathcal{Y}}_i$ **alors**
      $\boldsymbol{\theta}_t \leftarrow$ mise_à_jour$(\boldsymbol{\theta}_{t-1}, \mathbf{x}, i, \tilde{\mathcal{Y}}_i, y_i^*, h)$; `// éq. (3.7) ou (3.8)`
    fin
    empiler(y_i^*, h);
  fin
fin

La procédure d'apprentissage, décrite par l'algorithme 3, consiste à effectuer successivement l'inférence pour chaque séquence d'entrée et à corriger le vecteur de poids chaque fois qu'une décision erronée est prise. De manière cruciale (Wolpert, 1992; Ross et Bagnell, 2010), l'historique des décisions passées utilisé lors de l'apprentissage doit se composer des étiquettes jusque-là prédites, et non des

10. La complexité de l'apprentissage et de l'inférence ne dépend pas de la taille de l'historique, alors que pour les CRF, l'apprentissage et l'inférence ont une complexité qui croît exponentiellement avec l'ordre des dépendances, limitant en pratique l'utilisation de modèles d'ordre supérieur à 2.

étiquettes de référence comme c'est le cas dans le modèle original de (Daumé et Marcu, 2005), de manière à s'assurer que les données d'apprentissage reflètent réellement les conditions imparfaites qui seront observées lors du test. Cette modification est la principale différence entre l'algorithme 3 et la proposition initiale de (Daumé et Marcu, 2005).

L'utilisation d'un modèle à base d'historique, qui *réduit* un problème de séquence à une suite de problèmes de classification multi-classe, nous permet de généraliser cette approche au cadre ambigu, en nous inspirant du cadre théorique de Bordes *et al.* (2010) et Cour *et al.* (2011). Nous montrons comment il est possible d'adapter la règle de mise à jour du cas supervisé standard (§ 3.4.3.1) en fonction de l'information de supervision disponible, et ainsi d'étendre le modèle au cadre de l'apprentissage ambigu (§ 3.4.3.2).

3.4.3.1 Apprentissage (fortement) supervisé

Dans le cadre standard de l'apprentissage supervisé, l'étiquette correcte est connue pour chaque mot-occurrence : une prédiction est donc considérée comme incorrecte dès que l'étiquette prédite n'est pas celle de référence. Dans ce cas, une mise à jour de perceptron standard est appliquée :

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \phi\left(\mathbf{x}, i, y_i^*, h_i\right) + \phi\left(\mathbf{x}, i, \tilde{y}_i, h_i\right) \tag{3.7}$$

où y_i^* et $\tilde{y}_i$ sont respectivement les étiquettes prédites et l'étiquette de référence. Cette mise à jour peut être vue comme un pas de descente de gradient stochastique qui vise à augmenter le score de l'étiquette de référence et à diminuer celui de l'étiquette incorrectement prédite.

3.4.3.2 Apprentissage ambigu (ou faiblement supervisé)

Lors de l'apprentissage, chaque observation i est maintenant associée à un ensemble d'étiquettes possibles $\widetilde{\mathcal{Y}}_i$. Dans ce cas, nous choisissons de considérer une prédiction comme incorrecte lorsque l'étiquette prédite $\tilde{y}_i$ n'est pas l'une des étiquettes de référence, c'est-à-dire que $\tilde{y}_i \notin \widetilde{\mathcal{Y}}_i$ et l'on met alors à jour le vecteur de poids comme suit :

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \phi\left(\mathbf{x}, i, y_i^*, h_i\right) + \sum_{\tilde{y}_i \in \widetilde{\mathcal{Y}}_i} \phi\left(\mathbf{x}, i, \tilde{y}_i, h_i\right) \tag{3.8}$$

En comparant cette équation avec l'équation (3.7), on s'aperçoit que cette règle conduit à augmenter uniformément le score de toutes les étiquettes dans $\widetilde{\mathcal{Y}}_i$. Ce modèle sera noté MBH-A par la suite.

Dans le cadre de l'apprentissage ambigu Bordes *et al.* (2010); Cour *et al.* (2011), il est possible de montrer en faisant des hypothèses peu restrictives [11] qu'un classifieur entraîné uniquement à partir d'informations de supervision ambiguë revient à un classifieur appris à partir de l'information de supervision complète [12].

On peut donner une intuition sur un exemple simple : considérons un corpus contenant deux instances de la séquence « la souris » avec, respectivement, comme étiquetages (ambigus) [{DET} , {VERB, NOUN}] et [{DET} , {NOUN, ADJ}], et supposons que l'on a deux caractéristiques : une pour le mot courant et une pour le mot précédent. Après la première itération, la caractéristique binaire décrivant le mot précédent va être mise à jour deux fois lorsqu'elle est associée avec NOUN contre seulement une lorsqu'elle l'est avec une étiquette incorrecte (ADJ ou VERB). Grâce à ce partage de l'information, l'étiquette NOUN peut être maintenant correctement prédite pour « souris », même si cette configuration n'a jamais été vue lors de l'apprentissage de manière non ambiguë. De manière plus générale, tant que deux étiquettes ne sont pas systématiquement associées dans les ensembles de supervision, la répétition des mises à jour renforcera plus souvent la « bonne » étiquette et, au final, celle-ci finira par avoir le plus grand score.

3.5 Étude expérimentale

Dans cette section, nous explorons en détail les différentes configurations qu'il est possible de mettre en oeuvre et analysons les performances de nos méthodes. Nous effectuons également plusieurs expériences de contrôle qui permettent de comprendre dans quelle mesure il est possible d'apprendre en présence d'ambiguïté dans les données de supervision ainsi que l'importance relative des différentes contraintes. Enfin, nous analysons les erreurs résiduelles et montrons qu'une grande partie d'entre elles proviennent principalement de différences systématiques dans les conventions d'annotation. Ces erreurs ont donc peu de chances d'être traitées efficacement par de meilleures techniques d'apprentissage statistique. Nous pensons plutôt qu'une analyse linguistique plus précise des données pourrait être nécessaire afin d'uniformiser les différentes conventions, ce qui pourrait améliorer significativement les résultats et rendre les évaluations plus fiables et plus faciles à interpréter.

11. Qui reviennent à dire, en première approximation, qu'il suffit que l'étiquette correcte soit présente dans l'ensemble des étiquettes possibles et qu'elle n'y soit pas systématiquement associée à une autre étiquette

12. Bordes *et al.* (2010) définissent une fonction de perte dite *ambiguë (ambiguous loss)*, qui est optimisée par des mises à jour semblables à celles données par l'équation (3.8) et montrent que la solution qui permet d'obtenir l'erreur minimale pour cette fonction de perte est également la solution du problème minimisant la perte 0/1 que l'on pourrait évaluer si l'on connaissait l'étiquette de référence.

	Langue	Famille	Corpus parallèle	Provenance du test
ar	Arabe	Afro-Asiatique/Sémitique	NIST	Arabic Treebank (Maamouri et Bies, 2004)
cs	Tchèque	Indo-Européenne/Balto-Slave	EUROPARL	CoNLL'09 Shared Task (Hajič *et al.*, 2009)
de	Allemand	Indo-Européenne/Germanique	EUROPARL	UDT v2.0 (McDonald *et al.*, 2013)
el	Grec	Indo-Européenne/Hellenique	EUROPARL	Greek Treebank (Prokopidis *et al.*, 2005)
es	Espagnol	Indo-Européenne/Italiques	EUROPARL	UDT v2.0 (McDonald *et al.*, 2013)
fi	Finnois	Uralique/Fennique	EUROPARL	UDT v2.0 (McDonald *et al.*, 2013)
fr	Français	Indo-Européenne/Italique	EUROPARL	UDT v2.0 (McDonald *et al.*, 2013)
id	Indonésien	Austronésienne/Malayo-Polynésienne	Open Subtitle	UDT v2.0 (McDonald *et al.*, 2013)
it	Italien	Indo-Européenne/Italique	EUROPARL	UDT v2.0 (McDonald *et al.*, 2013)
sv	Suédois	Indo-Européenne/Germanique	EUROPARL	UDT v2.0 (McDonald *et al.*, 2013)

Tableau 3.3 – Descriptif des langues et des ressources utilisées pour les tâches de tranfert cross-lingue (chapitres 3 et 4).

3.5.1 Corpus et langues

Nous évaluons notre approche pour dix langues, de caractéristiques et de propriétés linguistiques variées, recouvrant différentes familles linguistiques, et qui sont listées au tableau 3.3. Pour toutes nos expériences nous utilisons l'anglais comme langue source. Pour évaluer les performances des méthodes proposées, nous avons utilisé plusieurs corpus arborés, ce qui nous permet d'obtenir un petit nombre de corpus annotés. Le tableau 3.3 regroupe les données que nous avons utilisées dans nos expériences. Les corpus de l'*Universal Dependency Treebank* sont directement annotés avec les étiquettes universelles de Petrov *et al.* (2012a). Pour les corpus d'autres provenances, ainsi que pour le côté anglais des corpus parallèles, les étiquettes morpho-syntaxiques sont simplifiées en étiquettes universelles en utilisant les correspondances préconisées par (Petrov *et al.*, 2012a).

Ces corpus ont été constitués manuellement par des experts linguistes et contiennent plusieurs types d'annotations, dont des étiquettes morpho-syntaxiques fines qui sont transformées en leur équivalent dans le jeu d'étiquettes universelles en utilisant les règles de Petrov *et al.* (2012a). La qualité des analyseurs entrainés est évaluée par leur taux d'erreur par occurrence sur le jeu de test.

3.5.2 Caractéristiques

Dans toutes nos expériences, nous utilisons des caractéristiques similaires à celles qui sont généralement utilisées dans des tâches d'analyse morpho-syntaxique :

- Pour le mot courant ainsi pour que les deux mots précédents et les deux mots suivants :
  - **identité du mot** : mot en minuscules s'il apparaît plus de 10 fois dans le corpus d'apprentissage ;
  - **suffixes** : les suffixes de 2 et 3 lettres s'ils apparaissent dans plus de 20 mots-types différents dans le corpus d'apprentissage ;
  - **classe** : la classe de ce mot [13] parmi 50 classes estimées sur le corpus d'apprentissage en utilisant MKCLS [14]. Les clusters de mots, appris de manière non supervisée, ont déjà été utilisés comme caractéristiques pour améliorer les performances de nombreuses tâches de TAL (Koo *et al.*, 2008; Täckström *et al.*, 2012; Owoputi *et al.*, 2013; Täckström *et al.*, 2013b) ;
- **majuscule** : une caractéristique binaire qui indique si le mot courant commence par une majuscule ou non ;

13. Les mots hors-vocabulaire lors du test sont arbitrairement associés à la classe 1.
14. http://code.google.com/p/giza-pp/

- **trait d'union** : une caractéristique binaire qui indique si le mot courant comporte un trait d'union ou non ;
- **type d'alphabet** : une caractéristique qui indique si le mot est écrit dans un alphabet grec ou latin ;
- **information de structure** : les étiquettes prédites pour les deux mots précédents (MBH-A seulement), la conjonction de ces deux étiquettes (MBH-A seulement), la conjonction de l'étiquette précédente et du mot précédent.

Ces caractéristiques sont semblables à celles qui sont utilisées dans Täckström *et al.* (2013b); Li *et al.* (2012b), exceptées les informations de structure qui ne peuvent être facilement considérées dans un modèle de séquence linéaire comme les CRF.

Comme unique prétraitement additionnel, nous normalisons tous les nombres (c'est-à-dire les mots constitués de seuls chiffres) par un unique mot spécial `__digits__`.

3.5.3 Conditions expérimentales

Pour chaque paire de langues considérée, les corpus parallèles sont alignés en utilisant la chaîne de traitement standard de Moses Koehn *et al.* (2007) avec l'heuristique d'intersection pour fusionner les deux directions d'alignement. Cette heuristique ne conserve que les liens prédits conjointement dans les deux directions, qui correspondent intuitivement aux alignements les plus sûrs.

Les étiquettes morpho-syntaxiques pour les phrases sources en anglais sont prédites automatiquement en utilisant Wapiti (Lavergne *et al.*, 2010b), un modèle CRF linéaire standard entraîné sur le Penn Treebank. Les étiquettes sont ensuite transférées vers les phrases cibles en utilisant la méthode décrite dans la partie 3.3. Pour toutes les langues, nous avons réextrait les contraintes de types avec nos propres outils [15].

Pour les deux modèles MBH-A et CRF-A, nous utilisons les contraintes de types au moment du décodage pour restreindre les étiquettes possibles pour un mot, mais conservons l'ensemble de toutes les étiquettes $\mathcal{Y}_i = \mathcal{U}$ lors de l'apprentissage. Nous justifierons ce choix au chapitre 4.

L'évaluation est effectuée sur les parties de test des différents corpus arborés, pour lesquelles les étiquettes de référence sont connues. Remarquons que les langues considérées ici ne sont pas réellement des langues peu dotées et qu'il est nécessaire d'avoir au moins un petit corpus pour pouvoir évaluer la qualité des approches. Nous utilisons comme métrique pour la tâche d'analyse morpho-syntaxique le taux d'erreur standard, c'est-à-dire le ratio d'occurrences dont l'étiquette prédite est erronée par rapport au nombre total d'occurrences.

15. Ces ressources sont disponibles à l'url `https://perso.limsi.fr/wisniews/weakly`

Hyperparamètres Pour toutes les paires de langues considérées, un analyseur morpho-syntaxique est entraîné à partir des étiquettes ambiguës (MBH-A). Le nombre d'itérations dans l'algorithme 3 est fixé à $T = 100\,000$, ce qui revient à dire que les paramètres de notre méthode sont estimés sur un sous-corpus de 100 000 phrases choisies aléatoirement dans le corpus d'apprentissage. Nos expériences préliminaires indiquent que le choix de ces phrases n'a que peu d'impact sur les performances obtenues. Il apparaît également que considérer plus de phrases ne permet pas d'améliorer les performances.

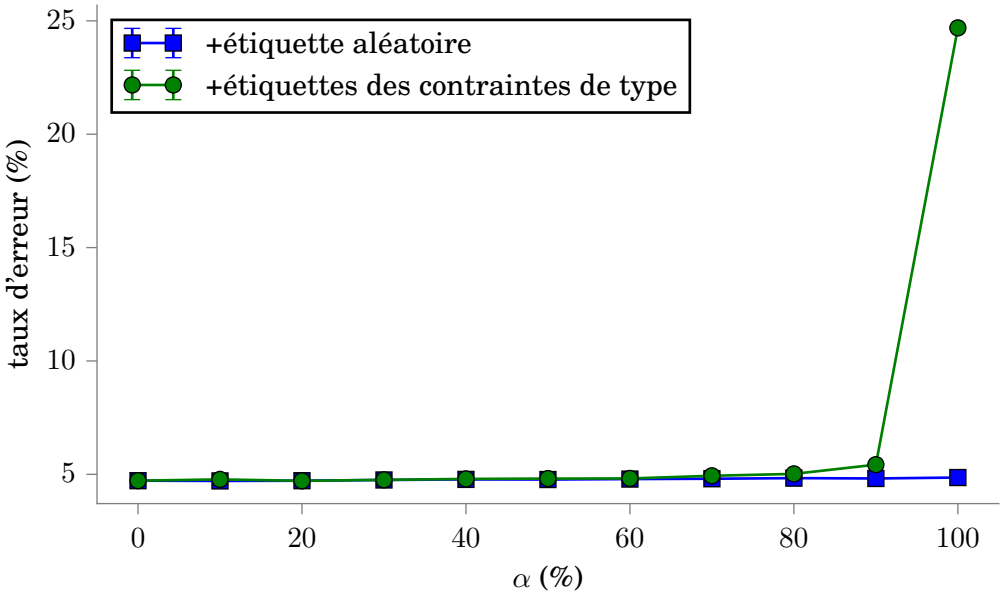
Nous donnons également les résultats pour une réimplémentation du modèle CRF partiellement observé (CRF-A) de Täckström *et al.* (2013b) avec le même jeu de caractéristiques que pour MBH-A— exception faite des caractéristiques d'ordre trois — , mais ne filtrons pas les formes possibles ni les suffixes en fonction des fréquences d'observation du corpus. Au lieu de cela, nous laissons le modèle sélectionner de lui-même les caractéristiques utiles en opérant une sélection de caractéristiques implicites grâce à la régularisation $\mathcal{L}_1$. Nous utilisons une régularisation $\mathcal{L}_1 + \mathcal{L}_2$, ce qui donne en pratique de meilleurs résultats que la régularisation $\mathcal{L}_2$ seule comme dans (Täckström *et al.*, 2013b). Pour chaque langue, nous choisissons aléatoirement 100 000 phrases [16] et effectuons 30 itérations [17] de l'algorithme de propagation résiliente (§ 2.5) avec la régularisation élastique ($\mathcal{L}_1 + \mathcal{L}_2$). Dans un contexte de langues cibles peu dotées en ressources, il n'est pas possible d'utiliser un corpus de développement pour choisir les hyperparamètres des modèles. Comme dans (Täckström *et al.*, 2013b), nous fixons arbitrairement les deux hyperparamètres de régularisation du CRF partiellement observé à 1.

3.5.4 Apprendre à partir d'exemples ambigus : une expérience de contrôle

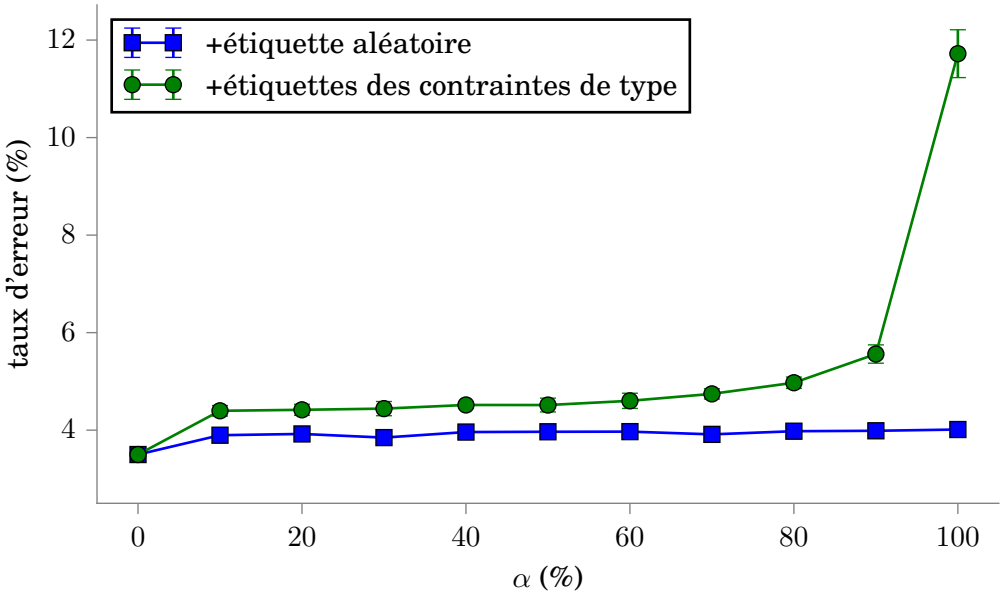
Afin de vérifier l'efficacité des méthodes présentées à la section 3.4, nous proposons deux expériences de contrôle. À partir du problème standard entièrement supervisé, nous ajoutons aléatoirement un certain nombre d'étiquettes fallacieuses de façon à contrôler l'ambiguïté du problème et ainsi tester la robustesse des méthodes lorsque que l'on considère le cadre de l'apprentissage ambigu. Comme nous ne disposons pas de corpus parallèles annotés, nous utilisons ici les parties de corpus d'apprentissage des différents corpus arborés 3.3. À partir des données de supervision, nous ajoutons des étiquettes fallacieuses de deux manières différentes.

16. Ceci afin de pouvoir comparer cette approche avec le modèle MBH-A où nous utilisons $T = 100\,000$ itérations au total. Il est possible d'utiliser l'intégralité du corpus, mais l'utilisation d'un plus grand nombre de phrases ne permet pas d'augmenter sensiblement les performances, résultat également remarqué dans (Täckström *et al.*, 2013b).

17. Ce qui suffit largement à atteindre la convergence.



(a) CRF-A



(b) MBH-A

FIGURE 3.4 – Évolution du taux d'erreur de MBH-A et CRF-A, sur le corpus de test français, en ajoutant dans le corpus d'apprentissage, pour $\alpha\%$ des positions, à l'étiquette de référence : (a) une étiquette fallacieuse aléatoire ; ou (b) l'ensemble des étiquettes autorisées par les contraintes de types pour le mot à cette position. Tous les points sont des moyennes de 10 expériences indépendantes, la déviation standard est également représentée (mais inférieure à la taille du point en général).

Premièrement, pour chaque position de la phrase étiquetée, avec une probabilité α, nous ajoutons une étiquette tirée au hasard parmi les autres étiquettes (c'est-à-dire $\mathcal{U} \setminus \{\tilde{y}_i\}$). Ainsi, lorsque $\alpha = 0$, on retrouve le cadre entièrement supervisé, et lorsque $\alpha = 1$, l'étiquetage de toutes les occurrences est ambigu avec exactement deux étiquettes. Comme le montre la figure 3.4 pour le français, augmenter α et donc le niveau d'ambiguïté n'a qu'un impact très limité sur les performances tant de MBH-A que de CRF-A. De plus, comme cela a déjà été remarqué précédemment (Dredze *et al.*, 2009), on s'aperçoit qu'ajouter un peu d'ambiguïté permet même dans certains cas de diminuer légèrement le taux d'erreur par rapport au cadre entièrement supervisé, probablement en améliorant la robustesse du modèle appris.

Cette expérience montre clairement que l'apprentissage reste possible en présence d'un étiquetage ambigu, du moins lorsque l'ambiguïté est aléatoire, comme le prédit la théorie (Bordes *et al.*, 2010; Cour *et al.*, 2011). Cependant, on peut remarquer que cette expérience ne décrit pas exactement la situation du transfert cross-lingue à laquelle nous nous intéressons. En effet, lorsque l'on utilise les contraintes de types, comme nous l'avons décrit à la section 3.3, chaque occurrence est soit entièrement désambiguïsée par les contraintes d'occurrences, soit associée aux contraintes de types et, dans ce cas, ce sont donc toujours les mêmes pour toutes les positions, pour un même mot-type. En d'autres termes, l'ambiguïté est toujours constituée des mêmes étiquettes et non d'une étiquette aléatoire parmi les étiquettes universelles. Pour mieux coller à cette situation, nous avons effectué une deuxième expérience de contrôle dans laquelle pour chaque position, toujours avec probabilité α, au lieu d'utiliser la seule étiquette de référence, nous utilisons toutes les étiquettes données par les contraintes de types pour ce mot-occurrence[18]. Pour $\alpha = 0$, on retrouve à nouveau le cadre entièrement supervisé ; pour $\alpha = 1$ en revanche, on obtient maintenant le cadre non-supervisé mais en présence de contraintes de types, ce qui correspond au cadre expérimental de (Das et Petrov, 2011; Li *et al.*, 2012b). Cette expérience peut également être vue comme une manière d'estimer l'impact des contraintes d'occurrences : à partir de la situation où chaque mot-occurrence est associé aux étiquettes des contraintes de types ($\alpha = 1$), les contraintes d'occurrences sont aléatoirement utilisées pour $(1 - \alpha)\%$ des occurrences (ce qui correspond au ratio de liens d'alignement dans notre cadre cross-lingue). Afin de séparer l'impact dû aux erreurs d'alignements et afin d'isoler le seul effet de l'ambiguïté, nous avons utilisé comme contraintes de types un dictionnaire calculé directement sur le corpus d'apprentissage du corpus arboré.

La figure 3.4 montre qu'augmenter α[19] n'a qu'un assez faible impact sur les

18. Dans les contraintes de types que nous utilisons pour cette expérience de contrôle, nous avons la garantie que l'étiquette de référence est autorisée par ces contraintes, cela revient donc à ajouter des étiquettes fallacieuses.

19. Et donc le niveau d'ambiguïté. Remarquons que lorsque l'on utilise les contraintes de types

performances de MBH-A and CRF-A, du moins lorsqu'une proportion suffisante d'occurrences sont intégralement désambiguïsées. Même lorsque seulement 20% des contraintes d'occurrences sont appliquées (i.e., $\alpha = 80\%$), il est encore possible d'apprendre un analyseur qui se comporte presque aussi bien que lorsqu'il est entraîné de manière supervisée. Cependant, lorsque les mots sont systématiquement associés avec le même ensemble d'étiquettes ($\alpha = 1$), le taux d'erreur augmente significativement. Dans ce cas, la méthode d'apprentissage décrite à la section 3.4.3 ne sera plus capable de désambiguïser l'information de supervision. Formellement, cela implique que les hypothèses sur lesquelles se basent les résultats théoriques de (Bordes *et al.*, 2010) et (Cour *et al.*, 2011) ne sont plus satisfaites.

Ce premier jeu d'expériences contrôlées nous permet de voir que les deux méthodes ici étudiées sont capables d'apprendre malgré une ambiguïté dans les données et montre clairement l'importance des contraintes d'occurrences. Lorsque seules les contraintes de types sont utilisées, ce qui correspond au cas $\alpha = 1$ dans la figure 3.4, toutes les occurrences sont systématiquement associées au même ensemble d'étiquettes et les possibilités d'apprentissage sont fortement réduites. Dans ce cas, le taux d'erreur pour le français est d'ailleurs relativement proche du taux de mots ambigus.

3.5.5 Quelles contraintes utiliser ?

contraintes	ar	cs	de	el	es	fi	fr	id	it	sv
∅	43.1	18.6	14.8	17.5	15.9	22.6	15.8	17.4	15.0	12.7
bitexte	42.2	21.2	16.1	17.0	13.2	25.3	14.3	18.1	13.8	13.4
wiki	45.0	15.9	14.9	12.8	13.4	20.9	17.3	22.5	12.7	13.5
wiki ∪ bitexte	25.3	17.3	12.7	16.0	12.3	16.5	13.1	14.0	12.7	11.1
wiki ∩ bitexte	25.1	10.7	8.6	8.4	8.2	11.1	9.8	11.1	9.1	9.6

Tableau 3.4 – Taux d'erreur (en %) pour MBH-A en variant les contraintes de types (∅ si aucune contrainte de types n'est utilisée). Les contraintes d'occurrences sont toujours utilisées. Les différentes contraintes de types sont ici appliquées à la fois pour obtenir les données de supervision et lors du test pour réduire l'espace des possibles.

pour certaines positions aléatoires, cela n'augmente pas nécessairement l'ambiguïté lorsque α augmente, du fait que certains mots peuvent être associés à une seule étiquette permise par ces contraintes.

contraintes	test	ar	cs	de	el	es	fi	fr	id	it	sv
∅		24.0	17.7	13.8	17.2	15.4	20.1	14.8	15.8	13.8	13.1
bitexte		23.3	17.3	13.6	17.0	14.8	19.2	14.3	14.8	13.5	12.4
wiki		24.7	7.8	9.5	8.3	11.4	12.6	9.8	11.2	9.5	9.7
wiki ∪ bitexte		23.3	17.3	13.3	16.8	14.7	19.2	14.1	14.8	13.3	12.5
wiki ∩ bitexte		23.9	8.3	9.7	8.4	11.2	12.7	10.0	11.1	9.4	9.5
∅	✓	23.9	17.6	12.4	17.0	15.2	19.6	14.5	15.6	13.7	13.0
bitexte	✓	27.0	17.3	12.3	17.5	14.4	18.1	14.9	15.0	13.3	12.8
wiki	✓	24.6	7.3	8.2	9.8	9.4	10.9	9.7	11.2	9.8	9.3
wiki ∪ bitexte	✓	27.0	16.7	11.8	16.3	12.4	17.4	13.7	14.6	12.7	12.0
wiki ∩ bitexte	✓	27.6	8.0	8.4	9.9	9.2	10.5	10.3	11.3	9.8	9.6

Tableau 3.5 – Taux d'erreur (en %) atteint par CRF-A en fonction des configurations de contraintes de types (contraintes), et selon que ces contraintes sont utilisées lors du test pour réduire les étiquettes possibles. Les contraintes d'occurrences et les contraintes de types sont systématiquement utilisées ici pour calculer les étiquettes de référence ambiguës.

Les tableaux 3.4 et 3.5 synthétisent les taux d'erreur obtenus par les deux méthodes que nous avons décrites à la section 3.4, en contrastant différentes combinaisons de contraintes.

En premier lieu, nous observons encore une fois de manière assez claire l'importance de considérer les deux sources de contraintes : contraintes d'occurrences et contraintes de types. Ne prendre en compte que les contraintes d'occurrences augmente, en moyenne, le taux d'erreur d'un facteur deux par rapport à la meilleure configuration. Nous observons également que prendre l'intersection des deux sources de contraintes de types (et donc de privilégier une ambiguïté réduite pour l'étiquetage automatique du corpus d'apprentissage) permet de réduire le taux d'erreur de manière importante, et ce particulièrement pour MBH-A.

MBH-A se comporte mieux pour les langues romanes (français, espagnol et italien), pour lesquelles les contraintes de types et d'occurrences apportent de nombreuses informations fiables de désambiguïsation. En revanche, pour le tchèque, l'allemand, ou le finnois, ses performances sont bien moindres que celles de CRF-A, ce qui peut être dû à un plus grand nombre de mots inconnus ou à des alignements moins fiables.

Il est intéressant de remarquer que CRF-A reste fort bon, voire meilleur, lorsque l'on ne considère que les contraintes de types données par Wiktionnaire, ce qui n'est plus le cas de MBH-A. Ceci peut être dû au fait que ce dernier ne met

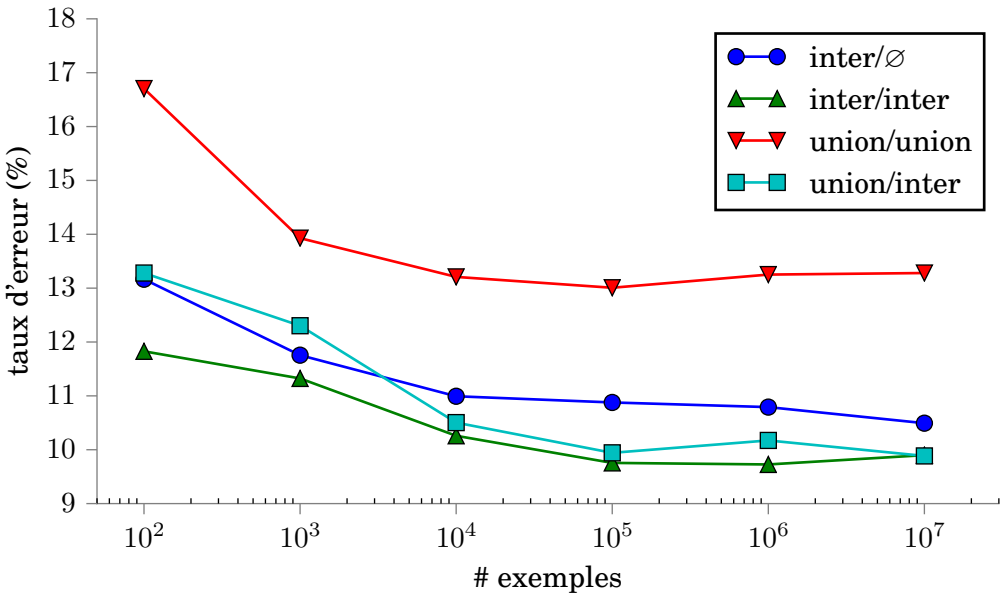


FIGURE 3.5 – Courbe d'apprentissage pour MBH-A pour le français en considérant différentes combinaisons possibles de contraintes, lorsque celles-ci sont utilisées pour construire les annotations ambiguës et/ou lorsqu'elles sont utilisées pour restreindre les prédictions possibles au test. La légende indique la configuration de contraintes utilisées lors de l'apprentissage/test : `inter` représente l'intersection des deux contraintes de types et `union` leur union.

à jour ses paramètres que lorsque l'étiquette prédite n'est pas autorisée par le dictionnaire, ce qui n'est pas si fréquent. Au contraire, CRF-A va toujours mettre à jour ses paramètres, même dans une situation partiellement informative. Le choix de contraintes plus strictes permet cependant à MBH-A d'obtenir de très bonnes performances, avec un coût computationnel moindre.

Afin de mieux comprendre l'importance des contraintes de types pour limiter les étiquettes possibles lors du test, nous avons représenté à la figure 3.5 les courbes d'apprentissage de MBH-A en fonction du nombre d'itérations, pour différentes combinaisons de contraintes de types à l'apprentissage (pour étiqueter automatiquement les instances) et au test (pour réduire l'ensemble des étiquettes possibles).

On peut tirer deux conclusions de ces courbes. Premièrement, on retrouve le fait qu'utiliser des contraintes de types lors du décodage permet d'améliorer les performances, mais encore faut-il que ces contraintes soient suffisamment strictes pour que leur utilisation apporte quelque chose. Deuxièmement, il semble que dans le choix des contraintes seul compte réellement celui effectué lors du test.

Utiliser l'heuristique d'union des contraintes lors du test donne des résultat très en dessous de ceux obtenus avec l'heuristique d'intersection, et cela quelle que soit l'heuristique de combinaison des contraintes de types utilisées pour compléter l'étiquetage par transfert cross-lingue.

Ces deux observations suggèrent que peu d'information est finalement capturée pendant l'apprentissage et que cette information est relativement redondante avec les contraintes de types. Cela implique que les capacités des modèles à prédire les parties du discours proviennent principalement de la réduction massive d'étiquetages possibles pendant le décodage.

Un dernier enseignement que l'on peut obtenir de la figure 3.5 est qu'à partir du moment où les contraintes de types sont suffisamment restrictives, un nombre restreint d'exemples suffit à entraîner un étiqueteur morpho-syntaxique. Ce résultat est particulièrement intéressant dans un contexte cross-lingue, dans lequel on est en droit de supposer que de gros corpus parallèles ne sont pas forcément disponibles entre une langue riche en annotations et une langue peu dotée. Dans cette configuration, 10 000 phrases (parallèles) suffisent pour atteindre la convergence, et un analyseur tout à fait correct peut être entraîné avec seulement 1 000 phrases [20].

3.5.6 Est-il si important de modéliser l'ambiguïté ?

Nous avons préalablement remarqué que l'ambiguïté des corpus d'apprentissage est assez faible, une fois que l'on applique à la fois les contraintes de types et les contraintes d'occurrences (voir le tableau 3.1), ce qui pose la question de l'importance de prendre réellement en compte ces occurrences ambiguës. Afin d'analyser cet effet, nous avons créé une variante pour chacune de nos méthodes d'apprentissage dans laquelle les positions ambiguës ne jouent plus aucun rôle pendant l'apprentissage et sont tout simplement ignorées.

Pour le CRF partiellement observé, on obtient simplement cela en remplaçant les ensembles $\widetilde{\mathcal{Y}}_i$ et $\mathcal{Y}_i$ par $\mathcal{U}$ dès que $\widetilde{\mathcal{Y}}_i$ contient strictement plus d'une étiquette. À l'apprentissage, les mots-occurrences sont donc soit entièrement désambiguïsés, soit complètement ambigus. Étant donné que le facteur de normalisation prend alors en compte toutes les étiquettes possibles, les positions entièrement ambiguës [21] ne jouent aucun rôle dans le calcul du gradient et ne vont donc aboutir à aucune mise à jour des paramètres. Ce modèle sera noté CRF-♯ par la suite.

Pour MBH-A, la généralisation du modèle à base d'historique au cas ambigu revenait à un algorithme d'apprentissage qui effectuait une mise à jour identique à celle d'un perceptron standard, sauf aux positions ambiguës où cette règle est adaptée. Une version ignorant les positions ambiguës revient donc à considérer la

20. À condition toutefois de disposer d'un dictionnaire de types assez riche.
21. C'était déjà le cas pour les positions où aucune contrainte ne s'appliquait.

règle de mise à jour suivante :

$$\boldsymbol{\theta}_{t+1} \leftarrow \begin{cases} \boldsymbol{\theta}_t - \phi\left(\mathbf{x}, i, y_i^*, h_i\right) + \phi\left(\mathbf{x}, i, \tilde{y}_i, h_i\right) & \text{if } |\widetilde{\mathcal{Y}}_i| = 1 \text{ and } \tilde{y}_i \neq y_i^* \\ \boldsymbol{\theta}_t & \text{sinon} \end{cases} \tag{3.9}$$

Cette mise à jour correspond à celle d'un perceptron standard entraîné uniquement sur les occurrences non ambiguës en ignorant les positions ambiguës. Ce modèle simple, que nous indiquerons par MBH-♯ par la suite, peut également être pris comme un modèle de base.

Pour évaluer l'impact des positions ambiguës à l'apprentissage — et ainsi justifier l'importance d'élaborer des méthodes plus complexes pouvant modéliser l'ambiguïté — nous proposons de comparer les versions générales de nos deux méthodes avec les variantes ignorant les positions ambiguës CRF-♯ et MBH-♯. Nous explorons à nouveau plusieurs configurations selon que l'on utilise, lors du test, les contraintes de types pour restreindre l'ensemble des étiquettes possibles ou non.

Les résultats de ces expériences pour les versions CRF-♯ et MBH-♯ sont présentés dans le tableau 3.6, où nous avons également reporté les résultats pour les méthodes générales CRF-A et MBH-A. Nous constatons que les résultats obtenus en ignorant les positions ambiguës pour MBH-♯ sont à peine moins bons que ceux que l'on obtient en les prenant en compte, *lorsque* les contraintes de types sont utilisées lors du décodage. Ceci confirme nos conclusions préalables (§ 3.5.5) relatives à MBH-A : une grande partie des résultats prédictifs de MBH-A provient principalement de l'effet des contraintes fortes lors de l'apprentissage et du décodage. Ces contraintes permettent à elles seules de désambiguïser la plus grande partie des occurrences, et ainsi de réduire considérablement l'ambiguïté. Pour l'exprimer autrement, les résultats obtenus par MBH-♯ montrent que les parties ainsi désambiguïsées suffisent pour apprendre un « bon » analyseur et que les occurrences qui restent ambiguës après le processus ne présentent qu'une source d'information supplémentaire marginale.

Quant à CRF-♯, les résultats alors obtenus sont pratiquement indistinguables de ceux de CRF-A, avec de plus une importance moindre sur l'utilisation ou non des contraintes de types lors du décodage, qui ne permettent de réduire le taux d'erreur que de manière mineure [22]. De manière intéressante, nous obtenons également pour les deux versions du CRF des résultats presque identiques lorsque l'on utilise uniquement les contraintes de types de WIKTIONNAIRE (Pécheux *et al.*, 2016b).

22. On observe même ici des cas où l'utilisation des contraintes de types au décodage diminue la performance, du fait des erreurs que les contraintes de types induisent alors.

	type	ar	cs	de	el	es	fi	fr	id	it	sv
MBH-A		35.9	14.6	11.7	9.2	10.3	20.2	10.8	18.5	10.7	13.1
MBH-A	✓	25.1	10.7	8.6	8.4	8.2	11.1	9.8	11.1	9.1	9.6
MBH-♯		36.4	15.3	11.7	9.7	11.7	20.5	10.7	19.3	11.4	13.2
MBH-♯	✓	25.4	10.7	9.6	8.5	9.4	11.1	10.1	11.3	9.2	9.7
CRF-A		23.9	8.3	9.7	8.4	11.2	12.7	10.0	11.1	9.4	9.5
CRF-A	✓	27.6	8.0	8.4	9.9	9.2	10.5	10.3	11.3	9.8	9.6
CRF-♯		24.1	8.7	9.7	8.3	11.3	12.6	10.2	11.0	9.5	9.5
CRF-♯	✓	27.7	8.3	8.4	9.8	9.4	10.4	10.4	11.3	9.8	9.5

Tableau 3.6 – Taux d'erreur (en %) pour MBH-♯ défini par l'équation (3.9) ainsi que pour CRF-♯, en fonction de l'utilisation ou non des contraintes de types au décodage (colonne 'type'). Pour toutes les expériences, les contraintes de types correspondent à l'intersection des dictionnaires.

	ar	cs	de	el	es	fi	fr	id	it	sv
MBH-A	25.1	10.7	8.6	8.4	8.2	11.1	9.8	11.1	9.1	9.6
MBH-S	8.5	1.5	5.0	—	2.4	5.9	3.5	4.8	2.8	3.8

Tableau 3.7 – Taux d'erreur (en %) pour le modèle à base d'historique, lorsque celui ci est entraîné de manière supervisée (MBH-S) ou entraîné dans le cadre ambigu (MBH-A) dans la meilleure configuration.

	ar	cs	de	el	es	fi	fr	id	it	sv
MBH-A	25.1	10.7	8.6	8.4	8.2	11.1	9.8	11.1	9.1	9.6
MBH-A + corrections	20.1	7.6	8.0	7.3	7.4	9.1	7.4	9.8	8.3	8.8

Tableau 3.8 – Taux d'erreur (en %) pour MBH-A lorsque certaines contraintes de types incorrectes ont été revues.

	ar	cs*	de	el*	es	fi	fr	id	it	sv*
(Li *et al.*, 2012a)	—	—	14.2	20.8	13.6	—	—	—	13.5	13.9
(Ganchev et Das, 2013b)	49.9	19.3	9.6	9.4	12.8	—	12.5	—	10.1	10.8
(Täckström *et al.*, 2013a)	—	18.9	9.5	10.5	10.9	—	11.6	—	10.2	11.1
CRF-A	24.6	7.3	8.2	9.8	9.4	10.9	9.7	11.2	9.8	9.3
MBH-A	25.1	10.7	8.6	8.4	8.2	11.1	9.8	11.1	9.1	9.6
(Duong *et al.*, 2014b)	—	—	7.5	7.9	8.4	—	—	—	10.1	—
MBH-A + corrections	20.1	7.6	8.0	7.3	7.4	9.1	7.4	9.8	8.3	8.8

Tableau 3.9 – Récapitulatif des taux d'erreur (en %) atteints par nos méthodes ainsi que celles de l'état de l'art. Les différents résultats ne sont directement comparables entre eux que si les ensembles de test sont les mêmes [25], ce qui est le cas pour le tchèque, le grec et le suédois, et qui est indiqué par un astérisque. Les modèles MBH-A et CRF-A correspondent aux configurations qui utilisent respectivement l'intersection des contraintes de types et WIKTIONNAIRE seul. Duong *et al.* (2014b) utilisent 1000 mots annotés et leur approche nous semble davantage comparable à notre modèle MBH-A pour lequel certaines erreurs d'annotations sont corrigées manuellement au sein des contraintes de types (+corrections ; voir § 3.5.9).

3.5.7 Amélioration par rapport à l'état de l'art

Le tableau 3.9 présente les meilleurs scores obtenus par les différents travaux sur le transfert cross-lingue et récapitule nos meilleurs résultats. Il est important de remarquer que la plupart des taux d'erreur ne peuvent pas être directement comparés entre eux, étant donné que tous les systèmes n'ont pas été entraînés ni évalués [23] sur les mêmes données (corpus [24], contraintes, alignements, etc.). Remarquons aussi que pour chaque référence, les résultats présentés sont ceux du meilleur système choisi directement et non sur un ensemble de validation.

On peut toutefois souligner que nos résultats dépassent les meilleurs chiffres jusqu'ici publiés, et ce pour toutes les langues considérées, et sont similaires à ceux de Duong *et al.* (2014a), qui utilisent des données manuellement annotées (1 000 occurrences).

23. La quasi-totalité des ressources utilisées dans les travaux antérieurs ne sont pas ou plus distribuées, ce qui complique fortement toute comparaison directe.

24. Les ensembles de test sont les mêmes pour le tchèque (cz), le grec (el) et le suédois uniquement (sv).

25. Duong *et al.* (2014b) utilisent la partie d'apprentissage des corpus arborés — sur lesquels ils ont au préalable prélevé 1000 mots — pour effectuer leurs tests. Leurs résultats ne sont donc pas directement comparables à ceux des autres travaux.

L'utilisation de ressources de meilleure qualité, notamment pour les contraintes de types qui ont été extraites d'une version plus récente de WIKTIONNAIRE pour la plupart des paires de langues, peut expliquer une partie des gains observés. Cependant, les résultats obtenus sur le grec, pour lequel nous utilisons les mêmes ressources que Li *et al.* (2012b), plus anciennes que celles utilisées dans Täckström *et al.* (2013b) et pour lequel le corpus de test est le même, montrent que ce n'est pas une justification suffisante.

3.5.8 Bilan

Pour résumer, les expériences décrites jusqu'ici nous permettent de mieux comprendre les raisons du succès de ces méthodes faiblement supervisées. Bien qu'il soit indiscutable que les méthodes que nous avons étudiées ici (et probablement d'autres) peuvent effectivement apprendre en présence de données partielles ou ambiguës, ceci n'arrive que lorsque l'ambiguïté est suffisamment aléatoire, ce qui permet aux étiquettes correctes de finalement émerger. Dans le scénario d'apprentissage que nous avons considéré ici, il nous semble que la raison principale qui permet à ces méthodes d'être efficaces est le fait que les contraintes d'occurrences, une fois filtrées, permettent de désambiguïser un nombre suffisamment important de positions de manière aléatoire, ce qui permet d'atteindre un niveau correct de performances.

3.5.9 Discussion

Les résultats que nous avons présentés dans ce chapitre montrent qu'un choix approprié des contraintes d'occurrences et de types pendant l'apprentissage et le décodage [26] est suffisant pour obtenir des performances qui dépassent largement celles des meilleurs systèmes de l'état de l'art. Cependant, cette évaluation quantitative ne donne que peu d'informations quant à l'utilité possible de ces méthodes. Dans cette section, nous donnons une interprétation plus pratique de ces résultats pour essayer de comprendre l'écart qu'il reste à combler entre l'apprentissage complètement supervisé et l'apprentissage faiblement supervisé. Nous nous limitons à analyser les erreurs de MBH-A, mais des expériences préliminaires montrent que des conclusions similaires s'appliquent à CRF-A.

Le tableau 3.7 montre que le taux d'erreur obtenu par MBH-A reste largement supérieur à celui du même modèle à base d'historique entraîné de manière supervisée sur le corpus d'apprentissage arboré, que nous noterons MBH-S (pour Modèle à Base d'Historique Supervisé). En première analyse, les résultats obtenus

26. Les contraintes à l'appentissage sont ici utilisées uniquement pour définir l'espace de supervision ambigu. Nous verrons leur utilisation pour filtrer également l'espace de recherche, comme c'est le cas au test, au chapitre 4 suivant.

par les méthodes de transfert semblent donc encore éloignés des performances des meilleurs étiqueteurs morpho-syntaxiques entraînés de manière supervisée. Pour l'espagnol, par exemple, disposer des données annotées permet de réduire le taux d'erreur d'un facteur 4. Cependant, il serait prématuré de conclure trop rapidement. Il faut en effet remarquer que l'évaluation dans le cadre proposé dans ce chapitre comporte un fort biais par rapport à celui des méthodes supervisées. En effet, dans la plupart des travaux sur l'étiquetage morpho-syntaxique, l'évaluation est réalisée sur des corpus du même domaine [27] que celui des corpus d'entraînement [28], comme c'est le cas ici pour MBH-S. Les méthodes exploitant un transfert bilingue reposent en revanche sur des corpus d'apprentissage parallèles, qui ne sont que plus ou moins proches du corpus de test. Contrairement à l'expérience supervisée, le cadre d'évaluation de MBH-A est donc pénalisé par une différence entre le type de corpus d'apprentissage et celui de test, ce qui a deux conséquences importantes. Premièrement, les corpus d'apprentissage (issus des corpus parallèles) et les corpus de test (ici des corpus arborés) n'appartiennent pas au même domaine et ne suivent pas nécessairement les mêmes conventions de normalisation et de découpage en unités lexicales élémentaires, ce qui est une première source évidente d'erreurs. En plus de la différence de domaine, les données de test utilisées exploitent une segmentation en mots qu'il n'est pas toujours aisé de reproduire à l'apprentissage et les conventions d'étiquetage ne sont pas nécessairement les mêmes que celles qui sont utilisées lors de l'apprentissage. Deuxièmement, et peut-être de manière encore plus importante, de nombreuses erreurs semblent être dues à des différences systématiques entre les schémas d'annotation du corpus de test et ceux utilisés lors de l'apprentissage (i.e. de la source en anglais des corpus parallèles d'une part et du WIKTIONNAIRE d'autre part). Si certaines de ces différences peuvent effectivement être linguistiquement fondées ou refléter des différences systématiques entre la structure des langues et leurs usages, de nombreuses erreurs nous semblent uniquement provenir de différences arbitraires entre les conventions d'annotation. Ce problème, également remarqué par Duong *et al.* (2014a), constitue l'une des principales motivations de leur travail. Si le premier problème n'a qu'un effet limité sur les performances (il ne concerne que des mots isolés et n'a donc pas d'impact systématique) le second soulève un problème plus fondamental de notre approche ou, du moins, de son évaluation.

En effet, l'étiquetage d'un corpus repose sur des conventions qui peuvent varier d'une campagne d'annotation à une autre. Si ces conventions ne sont pas les mêmes pour les corpus de test et d'apprentissage, les prédictions seront entachées d'erreurs

27. Il existe cependant tout un champ de recherche sur l'adaptation au domaine lorsque ce n'est pas le cas, voir par exemple (Daumé et Marcu, 2006).

28. C'est d'ailleurs l'une des hypothèses principales de l'apprentissage statistique : les instances d'apprentissage comme de test sont supposées être des échantillons i.i.d d'une même loi, bien que ce ne soit jamais exactement le cas en pratique (§ 2.2).

systématiques et l'estimation des performances sera biaisée. Par exemple, dans le corpus français issu du French Treebank utilisé lors de notre évaluation, les nombres sont étiquetés soit comme des déterminants (DET), comme par exemple dans le fragment « Christian Blanc, 44 ans » ou « un prêt de 25 millions de dollars », soit comme des adjectifs (ADJ), comme dans « Le Monde du 12 janvier » ou « à la page 23 ». Dans le Penn Treebank en revanche, sur lequel sont apprises les étiquettes de la langue source qui seront transférées sur la langue cible, les nombres sont systématiquement associés à l'étiquette NUM. Nous pensons que cette différence est davantage due à un choix de convention qu'à une réalité linguistique. On peut trouver bien d'autres exemples : en grec, les noms propres sont étiquetés soit comme X (lorsqu'ils se réfèrent dans les faits à un mot étranger *et* qu'ils ne sont pas translitérés) ou comme NOUN (dans tous les autres cas), alors que ce sont toujours des NOUN en anglais. En français et en grec, les contractions d'une préposition et d'un déterminant, comme « στο » (« σε το », que l'on peut traduire par 'à les') ou en français « aux » sont étiquetés comme des ADP dans le corpus arboré Universal Dependency Treebank, mais comme DET dans WIKTIONNAIRE, et sont en général alignés avec un déterminant dans les corpus parallèles.

Ce problème d'annotation est d'autant plus important dans des configurations comme la nôtre, qui : dépendent de nombreuses sources d'information (dans notre cas : WIKTIONNAIRE, le corpus parallèle et le corpus de test), établies indépendamment ; qui suivent des conventions d'annotation éventuellement contradictoires ; et dont la réduction des étiquettes vers les étiquettes universelles peut être entachée d'erreurs (Zhang *et al.*, 2012). Pour illustrer ceci, et mieux comprendre l'impact de la différence des corpus d'apprentissage et de test, nous avons effectué trois expériences supplémentaires.

3.5.9.1 Correction manuelle des contraintes de types

Nous avons conçu une expérience dans laquelle les contraintes de types ont été corrigées à la main pour les erreurs les plus fréquentes de MBH-A. Ces erreurs concernent en général des mots outils et nous semblent pouvoir être considérées comme des erreurs de conventions d'annotation plutôt que comme des « erreurs » de WIKTIONNAIRE ou du dictionnaire bitexte. Par exemple, les contraintes de types pour « du », « des », « au » et « aux » ont été « corrigées » de DET à ADP. On s'aperçoit sur le tableau 3.8 que des corrections simples des contraintes de types permettent des gains importants pour MBH-A, ce qui montre l'importance de l'impact des divergences entre les conventions d'annotation. Une observation similaire est à l'origine des travaux de Duong *et al.* (2014a) qui proposent d'utiliser un ensemble réduit de phrases annotées manuellement pour corriger automatiquement les différences d'annotation. Remarquons qu'à la différence de Duong *et al.* (2014a) nous n'envisageons pas ici l'utilisation de données manuelles, mais simu-

lons une condition dans laquelle les contraintes de types répondraient aux mêmes conventions d'annotation, la chaîne de traitement de notre cadre ambigu restant exactement la même.

3.5.9.2 Effet du changement de domaine d'apprentissage

Afin de mieux cerner le problème que peuvent poser les différences de domaine et de nature entre les corpus d'apprentissage et de test, nous proposons d'entraîner notre modèle à base d'historique de manière supervisée, mais sur les *mêmes* données que MBH-A lors de son apprentissage ambigu. Pour cela, il faudrait pouvoir entraîner un analyseur morpho-syntaxique de manière supervisée sur les données parallèles utilisées lors de l'apprentissage de notre méthode faiblement supervisée. Comme il n'existe pas, à notre connaissance, de données parallèles dont les deux côtés sont étiquetés en PdD, nous avons créé un tel corpus de manière artificielle en étiquetant automatiquement les phrases cibles du corpus parallèle à l'aide d'un analyseur en catégories morpho-syntaxiques. Les expériences décrites ci-dessous ont été effectuées pour l'espagnol.

Nous avons utilisé deux stratégies différentes pour obtenir de telles étiquettes supervisées sur le côté cible des corpus parallèles. Premièrement, notre modèle MBH-S est préalablement entraîné sur la partie apprentissage des corpus arborés. Dans cette expérience, l'analyseur supervisé est donc entraîné sur les mêmes données d'apprentissage que MBH-A dans le cas ambigu, mais les étiquettes (supervisées) suivent le même schéma d'annotation que le corpus de test. Le taux d'erreur est alors de 4.2% pour l'espagnol, c'est-à-dire presque deux fois plus que pour MBH-S, ce qui montre l'impact du changement de domaine.

Deuxièmement, nous avons utilisé un analyseur indépendant, FREELING [29], pour annoter les données d'apprentissage. Cette fois-ci, en plus d'une divergence de domaine, il y a une éventuelle différence dans les conventions d'annotation, qui ne sont pas les mêmes pour FREELING que pour les corpus arborés de notre travail. On obtient alors un taux d'erreur de 6.1%, à comparer avec les 8.2% atteints par MBH-A. Ces deux expériences montrent qu'une grande partie des erreurs de nos modèles peuvent être attribuées à un problème de concordance entre les corpus utilisés pour l'entraînement et le test ainsi qu'à des conventions d'annotation distinctes. Ceci n'est pas spécifique au cadre ambigu proprement dit, étant donné que des modèles entièrement supervisés seraient pénalisés de la même manière.

Il faut également considérer, dans l'analyse des performances obtenues, que les systèmes faiblement supervisés utilisant les contraintes de types lors du test sont fortement limités par l'exactitude de ces contraintes. Bien sûr, l'utilisation de contraintes lors du test n'est nullement obligatoire, mais nous avons vu que c'était

29. http://nlp.lsi.upc.edu/freeling/

cependant la configuration dans laquelle nous avions les meilleures performances. L'exactitude des contraintes de types dépend elle aussi largement des conventions d'annotation. Pour le français par exemple, le faible taux d'exactitude, lorsque les contraintes ne sont pas corrigées manuellement, s'explique principalement par un faible nombre de mots-types très fréquents (par exemple « au » ou « du ») dont les étiquettes morpho-syntaxiques diffèrent de manière systématique entre WIKTIONNAIRE et le corpus de test.

L'évaluation de notre méthode, et plus généralement des méthodes faiblement supervisées, pose donc de nombreux problèmes méthodologiques et se révèle sensible à de nombreux biais, qui sont pourtant rarement discutés. L'interprétation des résultats obtenus doit être faite avec précaution, en particulier lorsqu'il s'agit de les comparer avec des méthodes supervisées, et les résultats obtenus pourraient être bien meilleurs qu'ils ne le laissent penser.

3.6 Conclusions

Nous avons considéré dans ce chapitre le problème de l'apprentissage d'un analyseur morpho-syntaxique lorsque les étiquettes de supervision ne sont que partiellement connues, par exemple lorsque celles-ci sont automatiquement transférées à partir d'une langue source plus riche en annotations. En abordant ce problème sous l'angle de l'apprentissage ambigu, nous avons montré qu'il était possible d'étendre un modèle à base d'historique en adaptant une mise à jour de type perceptron au contexte faiblement supervisé. En considérant également une extension des modèles CRF, nous avons étudié ces deux méthodes capables d'apprendre à partir d'informations de supervision ambiguës, pour un grand nombre de langues. À condition de filtrer des contraintes d'occurrences qui peuvent être obtenues en projetant à travers des liens d'alignement les annotations à partir d'une langue bien dotée et en les complétant à l'aide de dictionnaires de types, il est possible d'obtenir suffisamment de données de faible ambiguïté pour pouvoir apprendre efficacement. Nous avons montré que ces techniques permettent d'obtenir des gains importants par rapport aux meilleurs résultats de l'état de l'art.

Les performances obtenues dans un cadre de transfert cross-lingue sont cependant le résultat d'un grand nombre de facteurs imbriqués, dont les importances relatives sont difficiles à mesurer : bruit dans l'étiquetage de la langue source ou dans les liens d'alignement, couverture et précision des différents dictionnaires, différences de domaines et de conventions d'annotation entre les corpus utilisés, divergences systématiques entre les langues, etc. En effectuant une analyse précise de nos résultats et de ceux de l'état de l'art, et en proposant plusieurs expériences de contrôle, nous avons pu identifier plusieurs critères importants pour que ces méthodes soient efficaces. Il semble que les méthodes ambiguës requièrent les condi-

tions suivantes : (a) des langues source et cible suffisamment proches pour que le transfert de PdD fasse sens et que les liens d'alignement soient sûrs — ce qui nécessite donc également des corpus parallèles suffisamment grands ; (b) des contraintes de types avec une couverture suffisamment importante, qui jouent un rôle essentiel, à la fois pour permettre un filtrage efficace des contraintes d'occurrences et pour compléter celles-ci, et qui, lors du test permettent de réduire les étiquettes possibles ; (c) une grande proportion d'occurrences entièrement désambiguïsée par les contraintes d'occurrences et de types. Même si ce cadre de transfert cross-lingue se justifie naturellement par le manque de données annotées pour des langues peu dotées, il ne semble pas évident que ces trois conditions seront effectivement rencontrées dans un véritable scénario. Le cas de l'arabe, avec peu d'alignements un à un et pour lequel le taux d'erreur reste relativement élevé, nous alerte à ce sujet, et semble suggérer qu'il faudrait être capable de mieux prendre en compte les différences morphologiques entre les langues source et cible.

Enfin, nous discutons des difficultés et des limites que pose l'évaluation de telles méthodes. Les différences de segmentation en unités, de divergences de domaine et de conventions entre différents corpus annotés peuvent largement biaiser les résultats. Pour les langues qui se comportent le mieux, nous avons remarqué qu'au moins une grande partie des erreurs — par rapport au cadre entièrement supervisé — peut être attribuée à des différences systématiques de conventions d'annotation, un problème qui doit donc être traité avec soin si l'on souhaite appliquer ce genre de techniques. En fait, notre analyse, confirmée par d'autre travaux (Duong *et al.*, 2014a; Tiedemann, 2014), montre qu'il y a probablement plus à gagner en se concentrant sur de meilleures ressources plutôt que sur les méthodes d'apprentissage à proprement parler. Il apparaît au final que la mise en œuvre et l'évaluation des méthodes de transfert nécessitent tout de même un effort conséquent et un minimum de connaissances des langues mises en jeu, ce qui conduit à en relativiser l'intérêt et ne garantit pas directement leur applicabilité telle quelle pour des langues réellement peu dotées.

Chapitre 4

Contraintes de types dans les modèles CRF

Sommaire

Lorsque l'on dispose de connaissances *a priori* sur les sorties possibles d'un problème d'étiquetage, il semble souhaitable d'inclure cette information à la fois lors du décodage et de l'apprentissage, que ce soit pour simplifier la tâche de modélisation et/ou accélérer les traitements. Lorsque l'on dispose d'un dictionnaire de types comme au chapitre 3 précédent, dont on a vu que l'utilisation est bénéfique lors du test, il semble naturel de vouloir utiliser ces contraintes également lors de l'apprentissage. Dans ce chapitre, nous montrons que cette intuition n'est pas toujours correcte et observons, de manière *a priori* paradoxale, que l'utilisation de contraintes à l'apprentissage peut dégrader sévèrement les performances, alors même que les contraintes sont toujours utiles lors du test. Nous étudions ce paradoxe, montrons que le manque de contraste induit par les contraintes peut entraîner une forme de surapprentissage et proposons quelques méthodes pour limi-

ter ce phénomène indésirable. Ce chapitre recouvre en partie des résultats publiés dans (Pécheux *et al.*, 2015).

4.1 Introduction

Nous avons vu dans le chapitre 2 que de nombreux problèmes de TAL peuvent être formalisés comme des problèmes d'étiquetage de séquences (§ 2.1), bénéficiant de ce fait de méthodes et de résultats établis en apprentissage automatique. Dans nombre de ces applications, par exemple pour la tâche d'analyse morpho-syntaxique étudiée en détail dans le chapitre 3, il est intéressant de pouvoir introduire, de manière implicite ou explicite, des connaissances linguistiques *a priori* sur les étiquetages possibles. On peut ainsi vouloir restreindre les sorties uniquement à celles qui sont compatibles avec les connaissances en question. Ces contraintes présentent en outre souvent l'avantage de permettre de réduire le temps de calcul, ce qui est particulièrement critique pour les problèmes pour lesquels le jeu d'étiquettes est grand. Ces contraintes peuvent être des contraintes formelles sur la forme de la sortie, par exemple, dans une tâche de segmentation utilisant un encodage BIO [1] des étiquettes, on peut vouloir imposer qu'une étiquette 'O' ne précède jamais une étiquette 'I'. Elles peuvent également provenir de connaissances linguistiques extérieures, par exemple des règles syntaxiques [2] ou encore de dictionnaires, cas que nous avons rencontré au chapitre 3. Dans de nombreuses applications, les contraintes peuvent se révéler nécessaires pour réduire l'ensemble des sorties possibles et ainsi rendre réalisable l'exécution de méthodes qui seraient trop coûteuses sinon. L'analyse morpho-syntaxique pour les langues à morphologie riche implique de prédire une étiquette parmi des ensembles comprenant des centaines, voire des milliers d'étiquettes morpho-syntaxiques possibles. Les problèmes de désambiguïsation associés sont donc à la fois difficiles et computationnellement extrêmement coûteux, au point de rendre inopérantes les méthodes standard (Mueller *et al.*, 2013) sauf recours à des heuristiques simplificatrices. Ce problème est encore plus prononcé dans le cas de la traduction automatique, tâche pour laquelle il est absolument indispensable de pouvoir filtrer les candidats cibles possibles d'un segment source. Enfin, le problème apparaît également lorsque l'on cherche à s'atteler de manière jointe à différentes tâches reliées pour aboutir à un unique modèle. Gahbiche-Braham *et al.* (2012) ont montré qu'apprendre et inférer simultanément la segmentation et les parties du discours (PdD) permet d'améliorer sensiblement les performances du prétraitement de l'arabe, par rapport au

1. Formellement, un segment étiqueté « intérieur » (I) doit être précédé soit de « début » (B), soit d'un autre 'I', mais jamais d'une étiquette « en dehors » (O).

2. Nous étudierons le cas de règles syntaxiques comme contraintes de réordonnancement aux chapitres 6 et 7.

fait d'enchaîner successivement ces deux étapes de manière indépendante. Dans un tel cadre, les étiquettes de la tâche jointe résultent alors du produit cartésien de l'ensemble des étiquettes de chaque tâche prise séparément, ce qui augmente considérablement leur nombre et limite ainsi les tâches qu'il est possible de traiter.

On s'intéresse plus particulièrement dans ce chapitre à l'introduction de contraintes lors de l'apprentissage d'un étiqueteur morpho-syntaxique. Pour cela, nous supposons disposer d'un dictionnaire associant à chaque mot un sous-ensemble des étiquettes possibles. Ce dictionnaire peut refléter une connaissance linguistique préalable, par exemple en étant extrait automatiquement de WIKITIONNAIRE ou encore être déduit des données d'apprentissage, comme dans le chapitre précédent. Sous l'hypothèse que ce dictionnaire est correct, il semble naturel de vouloir prendre cette information en compte afin, d'une part, d'accélérer l'apprentissage et l'inférence et, d'autre part, d'améliorer la qualité des prédictions. Nous avons vu que cela était effectivement le cas lorsque ces contraintes étaient utilisées lors du décodage dans la tâche d'analyse morpho-syntaxique par transfert cross-lingue au chapitre 3. Dans ce chapitre, nous nous intéressons de plus à la possibilité d'utiliser ces contraintes lors de l'*apprentissage*. En réduisant l'ensemble des étiquettes pouvant être prédites et donc la taille des espaces de recherche associés, les contraintes devraient permettre en un certain sens de simplifier la tâche du modèle. En effet, ce dernier peut alors concentrer son apprentissage sur la discrimination entre des hypothèses réalistes, en nombre réduit, plutôt que de considérer des configurations qui ne peuvent pas se produire.

Nous montrons que cette intuition n'est pas toujours correcte et qu'ajouter une telle information, même lorsqu'elle est pertinente et exacte, peut conduire à une dégradation de la capacité de généralisation du système.

Le point de départ de cette étude a été la tentative de reproduire les résultats de Täckström *et al.* (2013a) dans la tâche d'analyse morpho-syntaxique pour des langues cibles peu dotées présentées au chapitre 3. Nous avons déjà vu que dans cette approche le dictionnaire peut être utilisé pour restreindre l'espace de recherche : la liste des étiquettes possibles pour chaque mot peut alors être réduite à un ensemble d'alternatives (les étiquettes autorisées par le dictionnaire) bien plus restreint que l'ensemble des étiquettes définies dans le schéma d'annotation.

Le tableau 4.1 rassemble les taux d'erreur obtenus par le CRF du chapitre 3 avec les conditions expérimentales de la section 3.5.3. En comparant la première et la deuxième ligne, on retrouve le fait qu'il est intéressant d'ajouter de manière explicite les contraintes de dictionnaire lors du décodage : ceci oblige le modèle à choisir l'une des étiquettes possibles et permet ainsi d'éviter certaines erreurs. En revanche, de manière surprenante, la troisième ligne de ce tableau montre qu'introduire ces contraintes lors de l'apprentissage dégrade sévèrement les performances.

Nous sommes donc, en apparence, face à un double paradoxe : (a) inclure des

appr.	test	cs	de	el	es	fi	fr	id	it	sv
✗	✗	17.3	13.3	16.8	14.7	19.2	14.1	14.8	13.3	12.5
✗	✓	16.7	11.8	16.3	12.4	17.4	13.7	14.6	12.7	12.0
✓	✓	21.2	15.8	17.6	15.5	27.4	23.1	27.9	15.1	14.7

Tableau 4.1 – Une série de résultats surprenants : taux d'erreur (en %) pour 9 langues, obtenus par le modèle CRF partiellement observé sur la tâche d'analyse morpho-syntaxique par transfert cross-lingue à partir de l'anglais du chapitre 3, selon que l'on utilise les contraintes de types pour définir l'espace de recherche à l'apprentissage (appr.) et/ou au test (test). L'intégration de contraintes à l'apprentissage dégrade systématiquement les performances. Les contraintes de types sont obtenues en réalisant l'union d'un dictionnaire déduit des alignements et d'un dictionnaire extrait du WIKTIONNAIRE (voir la section 4.3.2 pour plus de détails sur les conditions expérimentales et les langues considérées).

contraintes pourtant informatives pénalise le modèle ; (b) reproduire des conditions similaires à l'entraînement et au test n'est pas la meilleure configuration.

La contribution de ce chapitre est d'apporter des explications à ce comportement inattendu, afin de pouvoir y remédier. En analysant théoriquement l'effet de l'inclusion des contraintes dans le modèle (§ 4.2), il est possible de mettre en lumière les relations complexes qui existent entre les contraintes, la régularisation et le surapprentissage. Les résultats expérimentaux présentés à la section 4.3 montrent en effet que l'introduction de contraintes peut entraîner une forme de surapprentissage de certaines caractéristiques, qu'il est possible d'éviter. En particulier, il semble important de limiter, lors de l'apprentissage, l'impact des contraintes afin de garder une forme de contraste.

4.2 Contraintes dans les modèles exponentiels

Étant donné le cadre de l'apprentissage structuré et les modèles exponentiels décrits au chapitre 2, nous allons maintenant introduire formellement la notion de contrainte dans les espaces de recherche.

4.2.1 Fonction de contrainte et espaces restreints

Nous modélisons les contraintes par la notion de *fonction de contrainte* :

$$c : \mathbf{x} \in \mathcal{X} \rightarrow \mathcal{Y}^c(\mathbf{x}) \subseteq \mathcal{Y}(\mathbf{x}).$$

Remarquons qu'ici, ces fonctions sont déterministes et ne font pas partie du modèle.

Exemple 4.2.1. *Dans le cas de l'analyse morpho-syntaxique, soit $t : \mathcal{V} \to 2^{\mathcal{T}}$ un dictionnaire associant à chaque mot un ensemble d'étiquettes, on considère la fonction « contrainte dictionnairique » suivante, que l'on note abusivement également t :*

$$t : \mathbf{x} = (x_1, \ldots, x_{|\mathbf{x}|}) \in \mathcal{X} \to \mathcal{Y}^t(\mathbf{x}) = t(x_1) \times t(x_2) \times \cdots \times t(x_{|\mathbf{x}|})$$

qui n'autorise que les séquences d'étiquettes respectant, pour chaque mot, les contraintes données par le dictionnaire.

4.2.2 Modèles exponentiels avec contraintes

On peut maintenant étendre les notations de la section 2.2.1 en prenant en compte des contraintes sur l'espace de recherche, données par une fonction de contrainte c :

$$\forall \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}^c(\mathbf{x}), \quad p^c_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathcal{Z}^c_{\boldsymbol{\theta}}(\mathbf{x})} \exp(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})), \tag{4.1}$$

où le terme de normalisation devient :

$$\mathcal{Z}^c_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}^c(\mathbf{x})} \exp(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})). \tag{4.2}$$

Les contraintes influencent uniquement le calcul de la fonction de partition dans le modèle exponentiel : tout se passe comme si les sorties impossibles selon les contraintes avaient une probabilité nulle.

À l'apprentissage, en notant a la fonction de contrainte utilisée (a pour contraintes d'apprentissage), la log-vraisemblance conditionnelle (équation (2.7)) s'écrit :

$$\ell^a(\theta, \mathcal{D}) = \sum_{i=1}^{N} \log p^a_\theta(\widehat{\mathcal{Y}}(\mathbf{x}_i)|\mathbf{x}_i). \tag{4.3}$$

où $\widehat{\mathcal{Y}}(\mathbf{x}_i)$ est l'ensemble des étiquettes de références de $\mathbf{x}_i$, ce qui comprend le cadre de l'apprentissage supervisé comme celui de l'apprentissage ambigu (chapitre 3).

Tout comme à l'apprentissage, si l'on a accès à une fonction de contrainte de bonne qualité, il peut être avantageux d'exploiter celle-ci pour réduire les candidats possibles lors du décodage, et ainsi diminuer les risques d'erreur tout en augmentant la vitesse d'inférence. En notant d cette fonction de contrainte (d pour contraintes de décodage), cela revient à considérer la règle de décision :

$$\mathbf{y}^* = f(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}^d(\mathbf{x})} p^d_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}). \tag{4.4}$$

Intuitivement, il semble préférable d'utiliser le même espace de recherche lors de l'apprentissage et du décodage, *i.e.* $a = d$, mais il est important de bien comprendre que rien ne l'impose. Nous avons d'ailleurs vu dans l'introduction de ce chapitre un exemple où il était préférable de ne pas considérer la même fonction de contraintes lors de l'apprentissage et lors du décodage (deuxième ligne du tableau 4.1). Remarquons que l'on pourrait même envisager de mettre des contraintes plus strictes à l'apprentissage que lors du décodage [3]. Une des questions soulevées par cette étude est de se demander comment choisir optimalement $\mathcal{Y}^a(\mathbf{x})$ pour l'apprentissage et $\mathcal{Y}^d(\mathbf{x})$ lors du décodage.

4.2.3 Contraintes comme caractéristiques

Il est intéressant de noter qu'il est possible de représenter explicitement les contraintes, jusqu'ici externes au modèle, comme des caractéristiques associées à des poids qui en dissuadent la violation. Soit c une fonction de contrainte, et supposons qu'il existe un ensemble de caractéristiques $I \subset 2^d$ permettant d'encoder exactement le complémentaire de l'espace de recherche donné par l'application des contraintes :

$$\forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{y} \in \mathcal{Y}(\mathbf{x}) \qquad \mathbf{y} \notin \mathcal{Y}^c(\mathbf{x}) \Leftrightarrow \exists k \in I, \boldsymbol{\phi}_k(\mathbf{x}, \mathbf{y}) \neq 0.$$

Il suffit alors de fixer le poids de toutes ces caractéristiques à $-\infty$ pour obtenir un modèle sans contraintes équivalent au modèle avec contraintes. Dans le cas des contraintes de types, on peut considérer l'ensemble des caractéristiques `(mot, étiquette)` qui ne figurent pas dans le dictionnaire [4]. Pour parvenir à cette représentation équivalente, il est nécessaire d'associer à ces caractéristiques un poids de $-\infty$, une valeur qu'il n'est pas possible d'atteindre dans la configuration sans contraintes du fait de la régularisation. L'utilisation explicite de contraintes revient donc, dans ce cas, à ignorer la régularisation pour une certaine classe de caractéristiques, ce qui peut donc conduire à du surapprentissage.

4.3 Expériences

Nous considérons dans cette section expérimentale deux tâches d'analyse morpho-syntaxique. En premier lieu, nous étudions en détail l'analyse morpho-syntaxique

3. Mais comme on peut s'y attendre, nous avons observé alors de très mauvaises performances, allant jusqu'à 90% d'erreurs.

4. Les caractéristiques `(mot, étiquette)` sont typiquement utilisées dans les modèles. Ceci montre que les modèles peuvent encoder implicitement des contraintes de types, en associant aux caractéristiques `(mot, étiquette)` qui ne sont pas observées un poids très négatif.

de l'allemand en considérant différents jeux de caractéristiques et d'étiquettes possibles, et en utilisant des contraintes extraites de différentes manières du corpus d'entraînement (section 4.3.1). Cette première tâche nous permet d'étudier les phénomènes de manière précise et contrôlée. En second lieu, nous revenons sur l'analyse morpho-syntaxique par transfert cross-lingue, dans laquelle les contraintes de types apparaissent naturellement, en considérant l'intégration de celles-ci lors de l'apprentissage. Dans cette seconde tâche, nous mesurons l'importance que peut prendre l'intégration des contraintes, si l'on souhaite obtenir des performances satisfaisantes.

4.3.1 Apprentissage supervisé pour l'analyse morpho-syntaxique

4.3.1.1 Conditions expérimentales

Corpus et tâches On considère dans cette section la tâche d'analyse morpho-syntaxique de l'allemand à partir de données annotées. Nous utilisons le corpus arboré TIGER [5] (Brants *et al.*, 2004), avec le même partitionnement que Fraser *et al.* (2013), qui consiste à exclure les 10 000 dernières phrases du corpus pour constituer avec les 5 000 premières de celles-ci le corpus de développement et avec les 5 000 dernières le corpus de test. Le corpus complet contient 50 472 phrases, soit 888 238 mots étiquetés avec leur catégorie morpho-syntaxique. Les étiquettes pour cette tâche sont structurées en différents champs : la catégorie syntaxique (PdD) pouvant prendre 54 valeurs possibles, ainsi que des traits morphologiques : cas, nombre, genre, personne, temps, mode ; pouvant prendre respectivement 4, 2, 3, 3, 2, 3 valeurs [6]. Ainsi, le mot *legendären* peut être étiqueté `cs=ADJ,cas=gen,num=sg,gen=masc,pers=✗,tmp=✗,mode=✗`. Sur les 1 373 étiquettes possibles, 619 sont observées sur le corpus d'apprentissage. Nous étudions les tâches consistant à prédire l'étiquette syntaxique (PdD) et l'étiquette complète (CMS), c'est-à-dire la combinaison des parties du discours et des traits morphologiques.

Modèle Nous utilisons un CRF linéaire avec différents jeux de caractéristiques qui sont décrits par la suite. Le maximum de log-vraisemblance de l'équation (4.3) est calculé en utilisant 30 itérations de l'algorithme de propagation résiliente (Riedmiller et Braun, 1993). Nous utilisons de plus une régularisation $\mathcal{L}_1$ et $\mathcal{L}_2$ dont les hyperparamètres sont choisis par *grid search*, pour chaque expérience, dans $\{0, 0.1, 1\} \times \{0, 0.1, 1\}$ de manière à maximiser les performances sur le corpus de développement. Différents choix des contraintes de types impliquent un nombre

5. http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.en.html

6. Ainsi que les valeurs « non applicable » et « ambigu » que l'on traite ici comme des catégories à part entière.

très variable de caractéristiques et choisir la régularisation adaptée à chaque configuration est important pour ne pas interpréter à tort des différences de résultats qui seraient dues à une régularisation inappropriée.

Contraintes de types Nous envisageons trois manières différentes d'obtenir des contraintes de types à partir du corpus annoté : « *corpus* », « *corrigées* » et « *oracle* ». Les contraintes de *corpus* sont obtenues en considérant, pour chaque mot-type, l'ensemble des étiquettes auxquelles il est associé dans le corpus *d'apprentissage*. Par exemple « amüsiert » a pour seule étiquette ADJ dans le corpus. Cette méthode délivre cependant un dictionnaire incomplet (les mots hors du vocabulaire du corpus d'apprentissage ne sont pas couverts) et incorrect (certains mots ambigus ont pu n'être observés qu'avec une seule étiquette sur les données d'apprentissage). C'est par exemple le cas de « amüsiert », qui apparaît également avec l'étiquette VERB en test. Afin d'étudier l'impact de ces deux problèmes sur les phénomènes étudiés, nous considérons deux conditions oracles, au sens où nous utilisons pour les définir les données de développement et de test. La première consiste à corriger le dictionnaire ainsi extrait : pour chaque mot dans le vocabulaire d'apprentissage, on s'assure que toutes les étiquettes observées en développement et en test sont bien incluses ; si ce n'est pas le cas, on les ajoute (contraintes « *corrigées* »). On associe donc à « amüsiert » les étiquettes ADJ et VERB. Dans la deuxième, on extrait les contraintes sur l'ensemble des données (de développement et de test, en plus de celles du corpus d'apprentissage) et non sur les seules données d'apprentissage (contraintes « *oracle* »). Cela revient également à considérer les contraintes corrigées auxquelles on a également ajouté les contraintes de types pour les mots hors du vocabulaire d'apprentissage.

Évaluation Les performances sont évaluées en utilisant le taux d'erreur standard (rapport du nombre d'occurrences incorrectes sur le nombre total d'occurrences) (global). Afin d'affiner davantage nos analyses, nous donnons également les taux d'erreur pour les mots inconnus (*mots hors vocabulaire*, MHV) et pour les mots connus (*mots dans vocabulaire*, MDV), et au sein de ces derniers, les taux pour les mots ambigus (c'est-à-dire observés dans le corpus d'apprentissage avec au moins deux étiquettes différentes) (amb) et non-ambigus (non-amb).

4.3.1.2 Modèle MaxEnt simple

Nous commençons par un modèle exponentiel très simple (MaxEnt) comprenant deux patrons de caractéristiques, le premier pouvant tester les associations `(mot, étiquette)` pour le mot et l'étiquette courante, et le second testant l'étiquette courante seule `(étiquette)`. Notons que, dans ce modèle, il n'y a pas de dépendance entre étiquettes.

contraintes			MaxEnt				
type	appr.	test	global	MDV	MHV	amb	non-amb
✗	✗	✗	10.7	6.6	49.8	10.9	1.5
corpus	✗	✓	10.7	6.6	49.8	10.9	1.5
	✓	✓	15.5	6.6	100.0	11.0	1.5
corr.	✗	✓	10.7	6.6	49.8	10.9	1.5
	✓	✓	15.4	6.5	100.0	11.0	1.1
oracle	✗	✓	6.1	6.6	1.0	10.9	1.5
	✓	✓	6.0	6.5	1.1	11.0	1.1

(a) MaxEnt

contraintes			CRF				
type	appr.	test	global	MDV	MHV	amb	non-amb
✗	✗	✗	6.0	2.9	35.5	4.2	1.4
corpus	✗	✓	6.0	2.9	35.0	4.2	1.5
	✓	✓	11.8	4.1	85.8	6.2	1.5
corr.	✗	✓	5.7	2.6	35.1	4.1	0.7
	✓	✓	11.4	3.7	85.0	6.2	0.7
oracle	✗	✓	2.3	2.5	0.5	3.9	0.8
	✓	✓	2.1	2.3	0.5	3.8	0.5

(b) CRF d'ordre 1 simple

contraintes			CRF+				
type	appr.	test	global	MDV	MHV	amb	non-amb
✗	✗	✗	2.9	2.2	9.4	3.0	1.2
corpus	✗	✓	2.9	2.3	8.8	3.0	1.5
	✓	✓	8.2	2.6	61.4	3.5	1.5
corr.	✗	✓	2.4	1.7	9.0	2.8	0.5
	✓	✓	7.7	2.1	61.6	3.4	0.5
oracle	✗	✓	1.6	1.7	0.4	2.8	0.4
	✓	✓	1.6	1.7	0.4	2.9	0.4

(c) CRF d'ordre 1 avec un jeu de caractéristiques riche

Tableau 4.2 – Taux d'erreur (%) pour les modèles MaxEnt et CRF entraînés de façon supervisée pour la tâche d'analyse morpho-syntaxique sur le corpus TIGER, en fonction de différentes contraintes considérées à l'apprentissage (appr.) et/ou au test (test) : aucunes (✗) ; contraintes de types extraites du corpus d'apprentissage (corpus) ; corrigées en utilisant le corpus de test (corr.) ; et complétées (oracle). Le taux d'erreur est donné en prenant en compte tous les mots (global) ; les mots dans le vocabulaire (MDV) ; les mots hors vocabulaire (MHV) ; les mots ambigus (amb) ; et les mots non ambigus (non-amb). (les notations sont détaillées dans le texte).

Les résultats obtenus par le modèle MaxEnt sont détaillés dans le tableau 4.2. Si, conformément à l'intuition (on prédit toujours l'étiquette la plus fréquente associée à un mot), ajouter les contraintes de types au test ne change rien aux résultats, on peut s'étonner de l'impact négatif obtenu lorsque celles-ci sont incluses lors de l'apprentissage. Une analyse plus précise des résultats montre que cette baisse de performances est entièrement imputable aux mots inconnus : lorsque les contraintes oracles (qui incluent les étiquettes de tous les mots du corpus de test) sont considérées, les mots hors-vocabulaire sont systématiquement bien reconnus et les performances avec et sans contraintes à l'apprentissage sont équivalentes.

Cette expérience suggère que le principal problème lié à l'introduction des contraintes de types à l'apprentissage est de désambiguïser abusivement de trop nombreuses occurrences. En effet, une grande majorité des mots-formes du corpus d'apprentissage ne présentent pas d'ambiguïté et sont donc complètement désambiguïsés par les contraintes de types. Ces occurrences ne contribuent donc plus au gradient ni à l'optimisation de l'équation (4.3), ce qui implique qu'aucun paramètre n'est mis à jour. Dans le cas présent, cela entraîne en particulier que les paramètres relatifs aux *a priori* des catégories ne sont plus calculés que sur les mots ambigus. Or, les mots inconnus sont souvent plus proches des mots rares, eux-mêmes le plus souvent non-ambigus. Ainsi, l'étiquette associée à la caractéristique ayant le plus fort poids en l'absence de contraintes à l'apprentissage est NOUN, correspondant à 50% des mots inconnus, alors que l'étiquette ayant le plus grand *a priori* en appliquant les contraintes lors de l'apprentissage devient APPRART[7], qui ne correspond à aucun mot inconnu. On retrouve le fait que filtrer des étiquettes équivaut à relâcher la régularisation sur certaines caractéristiques, ce qui peut conduire à sous-apprendre d'autres caractéristiques utiles, ici les caractéristiques relatives aux *a priori* des étiquettes.

4.3.1.3 Prédiction de la catégorie syntaxique avec un CRF

On considère maintenant un modèle CRF d'ordre 1 comportant un jeu de caractéristiques standard. Pour les mots courant, précédent et suivant, on considère : le mot en lettres minuscules, ses préfixes jusqu'à une taille de 5, ses suffixes jusqu'à une taille de 2, s'il est en majuscules, s'il contient un trait d'union, s'il ne contient que des nombres, s'il contient un chiffre, la forme obtenue en identifiant majuscules, minuscules et symboles, avec et sans répétitions (par exemple pour 'États-Unis' on a 'Xxxxx.Xxxx" et 'Xx.Xx). On considère également les associations : des mots courant et précédent ; et des mots courant et suivant. Toutes ces caractéristiques sont considérées conjointement avec chaque étiquette possible, ce

7. Préposition avec article.

à quoi on ajoute l'étiquette courante seule et les bigrammes associant l'étiquette courante et les étiquettes suivante et précédente.

Les résultats obtenus par ce modèle sont dans le tableau 4.2 et sont au niveau de l'état de l'art (Müller *et al.*, 2013). Comme pour le modèle MaxEnt, les mots hors-vocabulaire constituent une part importante des erreurs. À nouveau, l'ajout des contraintes de types apprises sur le corpus d'apprentissage n'améliore pas les performances. En effet, comme illustré à la section 4.2.3, les caractéristiques `(mot, étiquette)` permettent d'apprendre les mêmes contraintes de manière endogène au modèle : on voit ici que cela est fait sans erreur. On observe cependant que corriger les contraintes issues de l'apprentissage permet d'obtenir des gains substantiels (réduction des erreurs de 2.9% à 2.4%). Cependant, que l'on corrige ou non les contraintes, les utiliser lors de l'apprentissage multiplie le taux d'erreur par un facteur d'environ trois. Le résultat paradoxal observé à la section 4.1 n'est donc pas spécifique au cadre du transfert cross-lingue ou de l'apprentissage partiellement supervisé. Ici encore, la dégradation observée pour les MHV explique une grande partie de la baisse des performances. On observe toutefois également une dégradation pour les mots ambigus présents dans le vocabulaire d'apprentissage. Contrairement à l'intuition initiale, réduire les candidats possibles pour permettre au modèle de n'avoir à discriminer que les étiquettes plausibles n'apporte, dans cette expérience du moins, aucun avantage. De manière intéressante, le phénomène disparaît dans la condition « oracle ». Comme le modèle est le même pour ces contraintes et pour les contraintes corrigées (puisque la seule différence est la prise en charge des MHV au test), on en conclut que savoir désambiguïser correctement les mots inconnus permet également de mieux prédire des mots voisins connus mais ambigus.

Deux hypothèses, mutuellement non-exclusives, peuvent expliquer ces résultats. La première, déjà évoquée à la section 4.3.1.2, met l'accent sur les mots complètement désambiguïsés par les contraintes de types à l'apprentissage. En effet, ces mots sont alors ignorés, alors que leurs statistiques et surtout les caractéristiques qu'ils partagent avec d'autres occurrences pourraient être utiles à d'autres endroits. Une seconde hypothèse est que l'introduction de contraintes rend les conditions d'apprentissage et de tests différentes, puisqu'à l'apprentissage tous les mots sont connus, ce qui n'est pas le cas au test. Cette incohérence entre l'apprentissage et le test pourrait également contribuer à la dégradation des performances.

Pour tester ces deux hypothèses, nous avons effectué deux expériences de contrôle. La première essaie de résoudre le second problème en introduisant des mots inconnus lors de l'apprentissage. Les mots rares (c'est-à-dire de fréquence faible) ont souvent un comportement syntaxique proche des mots inconnus (Jurafsky et Martin, 2009, chap. 6). Nous proposons donc de ne pas utiliser les contraintes de types pour ces mots rares et de leur assigner, uniquement pour l'apprentissage,

l'ensemble des étiquettes possibles. Dans nos expériences, nous considérons qu'un mot est rare si sa fréquence d'apparition — le nombre d'occurrences — est inférieure ou égale à un (freq1), à cinq (freq5) ou à dix (freq10). Le tableau 4.3 montre que cette heuristique permet partiellement de résoudre le problème observé. Pour le modèle simple (MaxEnt), cette heuristique suffit à ramener le modèle avec contraintes de types à l'apprentissage au même niveau que le modèle sans contraintes. Pour les modèles CRF, plus riches en caractéristiques, la dégradation est faible pour les conditions freq5 et freq10. On observe encore une fois que l'amélioration des performances résulte principalement d'un meilleur traitement des mots inconnus.

Une des limites de l'approche précédente est que pour les mots rares, toutes les étiquettes sont considérées, ce qui reste problématique dans des tâches où cet ensemble est très grand. Comme la difficulté semble surtout provenir des mots complètement désambiguïsés à l'apprentissage, dans une deuxième expérience, nous ajoutons pour chaque mot complètement désambiguïsé un certain nombre d'étiquettes aléatoires de manière à s'assurer qu'il y a au moins i compétiteurs (min-i) au total (en comptant l'étiquette de référence) à chaque position. Les résultats présentés dans le tableau 4.3 montrent que les performances obtenues sont bien meilleures que lorsque l'on applique les contraintes de base, et légèrement moins bonnes que lorsque l'on autorise toutes les étiquettes pour les mots rares. Il est enfin possible de n'ajouter des étiquettes aléatoires que pour les mots *rares* (et désambiguïsés par les contraintes), mais il s'avère que cela détériore légèrement les résultats. Il semble donc que le nombre de concurrents joue également un rôle important pour les performances.

4.3.1.4 Impact du nombre d'alternatives

Dans cette section nous proposons une expérience de contrôle visant à comprendre l'impact d'un nombre de candidats différents à chaque position. En effet, dans le cas standard, il y a exactement le même nombre d'étiquettes, ici 54, pour toutes les positions. On pourrait penser que l'application des contraintes de types, en changeant la structure du problème, introduit un biais et complexifie ainsi l'apprentissage. Le tableau 4.4 montre que ce n'est pas le cas. Dans le cas où l'on n'applique pas de contraintes de types, c'est-à-dire le cas de base, nous avons introduit artificiellement une ou plusieurs étiquettes supplémentaires : (a) une même 55^{e} nouvelle étiquette à chaque position, ce qui revient à augmenter le jeu possible d'étiquettes par une nouvelle étiquette jamais observée comme étiquette de référence ; (b) une nouvelle étiquette différente pour chaque occurrence, ce qui ajoute un très grand nombre d'étiquettes et empêche le modèle d'utiliser un faible nombre de caractéristiques pour les discriminer ; (c) un sous-ensemble parmi 54 nouvelles étiquettes, dont un nombre aléatoire (uniforme entre 1 et 54) est introduit à chaque

contraintes	global	MDV	MHV	amb	non-amb
✗	2.9	2.3	8.8	3.0	1.5
freq10	3.0	2.3	9.3	3.1	1.5
freq5	3.0	2.4	9.4	3.1	1.5
freq1	3.2	2.4	11.2	3.1	1.5
min10	3.2	2.3	10.9	3.1	1.5
min4	3.3	2.3	12.8	3.0	1.5
min2	3.6	2.3	15.6	3.1	1.5
corpus	8.2	2.6	61.4	3.5	1.5

Tableau 4.3 – Taux d'erreur (en %) d'un CRF supervisé pour la tâche d'analyse morpho-syntaxique sur le corpus TIGER avec le jeu d'étiquettes syntaxiques seules (PdD), en considérant à l'apprentissage les contraintes de types : uniquement pour les mots de fréquence supérieure à 10 (freq10), 5 (freq5), 1 (freq1) ; en s'assurant que toute position comprend un minimum d'étiquettes (min10, min4, min2) ; ou pour tous (corpus). Lors du test on utilise systématiquement les contraintes *corpus*.

position, ce qui revient au cas de (a) avec un nombre variable d'étiquettes par position. Le tableau 4.4 montre que le modèle n'est pratiquement pas affecté par l'introduction de ces étiquettes négatives fallacieuses. Pour les cas (a) et (c), il lui suffit par exemple simplement de pondérer négativement les caractéristiques associées à ces nouvelles étiquettes. Si on pouvait dans l'ensemble s'attendre à ce résultat, on aurait pu cependant observer une différence du fait de la régularisation ou d'un biais possible. Ces expériences confirment quantitativement qu'il n'y a pas de différence.

Les résultats sont légèrement différents lorsque l'on considère les contraintes de types lors de l'apprentissage. En effet, ajouter une étiquette à chaque position [8] permet ici d'assurer une forme de contraste aux positions désambiguïsées par les contraintes de types, c'est-à-dire où celles-ci ne proposent qu'un seul candidat. Ajouter un même nouveau candidat à chaque position ne permet que très légèrement de diminuer la perte due à l'utilisation des contraintes de types lors de l'apprentissage. En effet, discriminer une seule étiquette, qui est toujours un exemple négatif, n'est pas très compliqué pour le modèle. En revanche, la configuration où

8. Il s'agit ici d'ajouter une étiquette dans l'espace (contraint) de *recherche*. Cette expérience ne doit donc pas être confondue avec l'expérience de contrôle de la section 3.5.4 qui consistait à ajouter une étiquette dans l'espace de *référence* et avec un espace de recherche complet.

contraintes	global	MDV	MHV	amb	non-amb
✗	2.9	2.3	8.8	3.0	1.5
✗ + 1 étiquette (≠)	3.0	2.4	9.1	3.5	1.3
✗ + 1 étiquette (=)	3.0	2.4	9.2	3.5	1.3
✗ + x étiquettes	3.0	2.4	9.3	3.5	1.3
corpus + 1 étiquette (≠)	4.2	2.4	22.0	3.5	1.3
corpus + 1 étiquette (=)	7.5	2.5	54.3	3.9	1.3
corpus	8.2	2.6	61.4	3.5	1.5

Tableau 4.4 – Taux d'erreur (en %) d'un CRF supervisé pour la tâche d'analyse morpho-syntaxique sur le corpus TIGER avec le jeu d'étiquettes syntaxiques seules (PdD), considérant à l'apprentissage les contraintes de types (corpus) ou non (✗), en ajoutant aux étiquettes possibles : une nouvelle même étiquette supplémentaire (+ 1 étiquette (≠)) ; une nouvelle étiquette unique pour chaque occurrence (+ 1 étiquette (=)) ; ou en ajoutant un nombre aléatoire de nouvelles étiquettes (+ x étiquettes) (voir le corps du texte pour davantage de détails). Le taux d'erreur est donné en prenant en compte tous les mots (global) ; les mots dans le vocabulaire (MDV) ; les mots hors-vocabulaire (MHV) ; les mots ambigus (amb) ; et les mots non-ambigus (non-amb). Les notations sont détaillées dans le texte.

l'on ajoute à chaque position une nouvelle étiquette permet d'améliorer sensiblement les performances par rapport au cas de l'utilisation directe des contraintes de types. En effet, aux positions qui seraient entièrement désambiguïsées par les contraintes seules, et qui ne nécessitaient donc aucun effort d'apprentissage, il est maintenant nécessaire d'effectuer un travail non négligeable pour prédire la bonne étiquette, en mettant à jour les paramètres du modèle et en apprennant donc une information à ces positions. On retrouve donc les observations et les conclusions de la section précédente pour la configuration `min2`, avec un taux d'erreur qui reste cependant supérieur. Il vaut donc mieux assurer à chaque position un minimum de contraste avec des étiquettes qui peuvent apparaître comme étiquettes de références plutôt qu'avec des étiquettes arbitraires.

4.3.1.5 Analyse morpho-syntaxique pour le jeu d'étiquettes complet

Nous considérons ensuite la tâche de prédiction de l'étiquette morpho-syntaxique complète. Les étiquettes morpho-syntaxiques sont structurées, au sens où les traits morphologiques possibles dépendent de la catégorie syntaxique. Une approche possible est donc de découpler le problème en apprenant d'abord les étiquettes syntaxiques, puis en utilisant celles-ci pour filtrer les traits morphologiques (Müller

contraintes	global	MDV	MHV	amb	non-amb
freq1	14.4	12.1	37.2	14.0	6.3
min10	16.6	13.5	45.7	14.9	9.5
min4	17.5	13.7	53.4	15.2	9.5
min2	18.1	13.8	58.6	15.3	9.4
corpus	19.9	14.1	74.7	15.8	9.3

Tableau 4.5 – Taux d'erreur (en %) d'un CRF supervisé pour la tâche d'analyse morpho-syntaxique sur le corpus TIGER avec le jeu d'étiquettes syntaxiques complet (PdD + CMS), en considérant à l'apprentissage les contraintes de types : uniquement pour les mots de fréquence supérieur à 1 (freq1) ; en s'assurant que toute position comprend un minimum d'étiquettes (min10, min4, min2) ; ou pour tous (corpus). Lors du test on utilise systématiquement les contraintes *corpus*. Pour la tâche d'analyse morpho-syntaxique complète, il n'est pas possible de faire les expériences en un temps raisonnable pour les configurations freq10 ; freq5 ; et sans utiliser de contraintes de types.

et al., 2013). Dans ce travail, nous nous intéressons uniquement à la prédiction de l'étiquette morpho-syntaxique complète.

Pour cette tâche, le nombre total d'étiquettes rend prohibitive l'utilisation du modèle CRF précédent en l'état, et diverses heuristiques doivent être envisagées pour réduire l'espace de recherche. Il est de plus nécessaire de limiter le nombre d'étiquettes candidates pour les mots inconnus. Une première approche consiste à se limiter à l'ensemble des étiquettes observées (619 au lieu de 1373), ou bien encore aux étiquettes dites « ouvertes » [9], ce qui limite les étiquettes possibles à 435, ou enfin, selon l'approche retenue dans ce travail, de ne prendre en compte que celles qui sont observées avec des mots rares (de fréquence 1), ce qui réduit ce nombre à 204. Nous considérons les mêmes caractéristiques que pour le modèle de la section 4.3.1.3, à ceci près que chaque fois que l'on considérait une caractéristique portant sur une étiquette (catégorie syntaxique), nous considérons maintenant à la fois l'étiquette complète, la catégorie syntaxique et les combinaisons impliquant la catégorie syntaxique et chacune des catégories morphologiques. On aura donc, par exemple, une caractéristique testant à la fois la catégorie syntaxique et le cas des étiquettes courante et précédente. À notre connaissance, seuls Müller *et al.* (2013) et Silfverberg *et al.* (2014) ont également utilisé des caractéristiques internes aux étiquettes.

9. Estimées en partitionnant les données d'apprentissage et en imposant que la fréquence à laquelle une étiquette est vue avec un nouveau mot soit supérieure à un seuil (e.g. 10^{-4}).

En utilisant les contraintes de types, il est possible d'entraîner un modèle CRF standard sans avoir besoin d'utiliser d'autres heuristiques (contrairement, par exemple, à Müller *et al.* (2013)). Les résultats obtenus, dans le tableau 4.5, montrent que l'on retrouve le même comportement que pour la tâche d'analyse syntaxique simple. Garantir un minimum de compétiteurs à chaque position permet de choisir un bon compromis entre vitesse d'apprentissage et performances en généralisation. Imposer un minimum de 10 alternatives (min10) permet en effet un entraînement un peu plus de dix fois plus rapide qu'en omettant les contraintes pour les mots rares (freq1). On peut imaginer augmenter encore les performances au prix d'un entraînement plus long. De meilleures techniques qui permettraient de limiter les dégradations dues à l'introduction de contraintes de types, tout en conservant leur bénéfice computationnel, restent à trouver.

4.3.2 Apprentissage ambigu pour l'analyse morpho-syntaxique

La tâche d'analyse morpho-syntaxique de la section 4.3.1 nous a permis d'étudier le problème dans un cadre bien contrôlé. Dans ce cadre, les contraintes de types, même lorsqu'elles ne sont utilisées qu'au décodage, ne permettent jamais d'améliorer les performances. Il s'avère en fait que le modèle est capable de les apprendre presque parfaitement, et leur seul intérêt provient du gain important en vitesse qu'elles permettent. Ces contraintes étant exclusivement extraites du corpus d'apprentissage lui-même, elles n'apportent toutefois aucune information nouvelle, et peuvent donc être responsables du surapprentissage observé. Nous considérons ici un autre exemple, dans lequel les contraintes de types apparaissent de manière naturelle, sont extraites indépendamment du corpus d'apprentissage et se révèlent utiles pour améliorer les performances.

On considère la tâche d'analyse morpho-syntaxique faiblement supervisée introduite au chapitre précédent. Pour chaque langue, nous utilisons les deux sources de contraintes de types : d'une part un dictionnaire automatiquement extrait de Wiktionnaire et d'autre part des contraintes extraites du corpus d'apprentissage, annoté indirectement à partir de l'anglais à travers des liens d'alignement. Les contraintes extraites des bitextes jouent un rôle analogue aux contraintes extraites des corpus de la section 4.3.1, alors que les contraintes issues du Wiktionnaire reflètent une connaissance linguistique externe que l'on souhaiterait exploiter. En plus d'être utilisées pour apprendre la référence ambiguë, c'est-à-dire pour construire $\mathcal{Y}^r(\mathbf{x})$, les contraintes de types c peuvent restreindre l'espace de recherche $\mathcal{Y}^c(\mathbf{x})$ à l'apprentissage et au décodage, configuration retenue par Täckström *et al.* (2013a) et Wisniewski *et al.* (2014a). Les tableaux 4.1 et 4.6 montrent pourtant que comme pour l'apprentissage supervisé, inclure les contraintes à l'ap-

contraintes	appr.	test	cs	de	el	es	fi	fr	id	it	sv
bitexte	✗	✗	17.3	13.6	17.0	14.8	19.2	14.3	14.8	13.5	12.4
	✗	✓	17.3	12.3	17.5	14.4	18.1	14.9	15.0	13.3	12.8
	⊕	✓	17.2	12.4	18.3	14.7	18.8	18.6	16.0	13.4	13.3
	✓	✓	23.3	17.2	23.8	19.9	34.3	24.9	30.2	15.2	19.4
wiki	✗	✗	7.8	9.5	8.3	11.4	12.6	9.8	11.2	9.5	9.7
	✗	✓	7.3	8.2	9.8	9.4	10.9	9.7	11.2	9.8	9.3
	⊕	✓	7.3	9.0	14.5	9.8	11.4	9.6	12.2	12.4	9.6
	✓	✓	8.8	10.7	16.9	10.3	12.1	10.9	13.9	13.4	10.1
wiki ∩ bitexte	✗	✗	8.3	9.7	8.4	11.2	12.7	10.0	11.1	9.4	9.5
	✗	✓	8.0	8.4	9.9	9.2	10.5	10.3	11.3	9.8	9.6
	⊕	✓	8.0	8.8	12.6	9.3	11.4	11.9	11.9	10.8	9.7
	✓	✓	12.8	13.2	14.0	12.0	22.4	14.7	20.5	14.7	14.6

Tableau 4.6 – Taux d'erreur (%) obtenus par un modèle CRF partiellement observé sur la tâche d'analyse morpho-syntaxique par transfert cross-lingue (PdD seuls), selon que l'on utilise : aucune contrainte (✗) ; les contraintes de types uniquement lorsqu'elles sont différentes des contraintes de référence ⊕ ; ou les contraintes de types pour tous les mots (✓) pour définir l'espace de recherche à l'apprentissage (appr.) et/ou au test (test). Les contraintes de types sont obtenues en combinant un dictionnaire tiré des bitextes (bitexte) et un dictionnaire issu de WIKTIONNAIRE (wiki) (voir le tableau 4.1 pour le cas de l'union).

prentissage nuit aux performances, avec dans certains cas une différence très importante, par exemple pour le finnois (fi) ou l'indonésien (id) pour lesquels le taux d'erreur est quasiment doublé. En omettant les contraintes de types lors de l'apprentissage, ce simple changement permet d'obtenir un gain moyen de 6.0% sur les langues considérées par rapport au modèle [10] état de l'art de Täckström *et al.*

10. En réalité, à cause d'une erreur d'implémentation, les résultats publiés dans Täckström *et al.* (2013a) correspondent au cas où l'on n'applique aucune contrainte de types pour réduire l'espace de recherche (premières lignes de chaque bloc dans le tableau 4.6). Les résultats corrigés ont été publiés par la suite dans un *erratum* disponible ici `http://www.dipanjandas.com/files/erratum.pdf`. Bien que les auteurs considèrent que les résultats des deux configurations sont semblables, leurs résultats montrent pourtant clairement une différence de l'ordre de 2% en moyenne pour le cas des contraintes bitextes seules et de l'union, mais pas dans le cas des contraintes issues de WIKTIONNAIRE seules. Nous observons dans nos expériences une différence pour ce dernier cas également, que nous attribuons à la manière dont nous avons extrait les dic-

contraintes		taux d'erreur (%)								
appr.	test	cs	de	el	es	fi	fr	id	it	sv
✗	✗	12.1	9.9	9.0	12.2	14.5	10.7	12.0	10.2	11.2
✗	✓	10.1	9.2	8.2	9.7	10.7	10.1	11.0	9.1	9.9
✓	✓	10.5	10.1	8.7	9.6	11.6	10.1	11.9	9.2	10.0

Tableau 4.7 – Taux d'erreur obtenus par notre modèle à base d'historique MBH-A sur la tâche d'analyse morpho-syntaxique par transfert cross-lingue, selon que l'on utilise les contraintes de types pour définir l'espace de recherche à l'apprentissage (appr.) et/ou au test (test). Les contraintes de types sont obtenues par intersection d'un dictionnaire tiré des bitextes et d'un dictionnaire issu de WIKTIONNAIRE.

(2013a), ce qui montre l'importance de prendre en compte le problème.

La section 4.3.1 permet de comprendre en quoi les contraintes de types posent problème lors de l'apprentissage. En effet, pour toutes les positions sans lien d'alignement, les étiquettes de référence sont les mêmes que les étiquettes possibles : $\mathcal{Y}^r(\mathbf{x}) = \mathcal{Y}^a(\mathbf{x})$ et donc $p^a_\theta(\mathcal{Y}^r(\mathbf{x})|\mathbf{x}) = 1$ (voir l'équation (4.3)). Le modèle n'apprend donc rien pour cette position. Dans la figure 3.1, par exemple, le premier mot « Un » est associé à quatre étiquettes de référence, qui sont aussi les étiquettes possibles lorsque les contraintes de types sont appliquées à l'apprentissage. Cette observation est toujours vraie si les contraintes de types permettent de désambiguïser complètement un mot. Utiliser les contraintes de types revient donc à ignorer une grande partie des exemples. Une solution possible est alors d'utiliser à l'apprentissage les contraintes de types uniquement si les étiquettes ainsi restreintes sont strictement plus nombreuses que les étiquettes de référence. Par exemple, dans la figure 3.1, pour la première position, « Un » possède quatre étiquettes de référence possibles ; on utilise donc l'ensemble des douze étiquettes possibles pour définir l'espace de recherche. En revanche, pour la deuxième position, 'marché', seule l'étiquette NOUN est référence, on peut donc utiliser les contraintes de types et laisser le modèle apprendre à préférer NOUN à VERB uniquement.

Cette stratégie, indiquée par le symbole $\oplus$ dans le tableau 4.6, ne permet en fait pas d'améliorer les performances. Au contraire, elle les dégrade pour plusieurs langues. Il semble donc que, bien que les contraintes de types permettent d'accélérer considérablement la vitesse d'apprentissage (d'un facteur 15 environ), elles ne permettent pas de simplifier la tâche du modèle, et même, au contraire, dégradent à nouveau les résultats.

tionnaires en utilisant toutes les informations de forme de WIKTIONNAIRE, et donc en obtenant des contraintes de types plus puissantes (et donc plus dangereuses à l'apprentissage).

Remarquons enfin que l'impact négatif des contraintes lors de l'apprentissage ne concerne pas seulement les CRF : nous observons le même comportement pour notre modèle à base d'historique, MBH-A, décrit à la section 3.4.3 dont la mise à jour est semblable à celle d'un perceptron, entraîné dans les mêmes conditions, même si l'effet est ici bien moins important, comme le montrent les résultats du tableau 4.7.

Ce qui pourrait sembler un détail d'implémentation se révèle donc être, dans ce cadre, d'une importance capitale pour obtenir de bonnes performances. Contrairement à Täckström *et al.* (2013b) nous préconisons de ne pas utiliser les contraintes de types lors de l'apprentissage, ou alors de garantir un minimum de contraste comme nous l'avons montré dans nos expériences. Cette différence nous a permis d'améliorer de manière significative les meilleurs résultats publiés jusqu'ici.

4.4 Lien avec l'état de l'art

L'utilisation de contraintes de types pour l'analyse morpho-syntaxique a surtout été proposée dans le contexte de l'apprentissage non-supervisé (Mérialdo, 1994), que ces contraintes soient extraites de corpus (Goldberg *et al.*, 2008; Ravi et Knight, 2009; Naseem *et al.*, 2009) ou issues de ressources externes comme Wiktionnaire (Li *et al.*, 2012b).

Dans le cadre de l'apprentissage supervisé, filtrer les candidats possibles lors du décodage pour accélérer la vitesse de l'analyse morpho-syntaxique est une pratique standard (Ratnaparkhi, 1996; Moore, 2014). Hajic (2000) considère différentes manières d'obtenir des dictionnaires de types pour l'analyse morpho-syntaxique, similaires à nos conditions « corpus », « complétées » et « oracle », et constate que l'utilisation de ces dernières permet d'obtenir davantage de gains que n'en procure un accroissement des données d'apprentissage. Plus récemment, Moore (2014) propose une forme de lissage inspirée du lissage Kneyser-Ney utilisé pour les modèles de langues (Chen et Goodman, 1998), qui permet d'augmenter le rappel des contraintes extraites du corpus, et donc de se rapprocher de ce que nous avons appelé les contraintes « corrigées ». À notre connaissance, seuls Smith *et al.* (2005), Waszczuk (2012) et Östling (2013) font état de l'utilisation explicite des contraintes lors de l'apprentissage supervisé. Östling (2013) utilise une condition qui serait semblable à ce que nous aurions appelé `freq3`.

Smith *et al.* (2005) et Waszczuk (2012) séparent en deux étapes l'analyse morpho-syntaxique pour un jeu d'étiquettes complexe : une étape de *proposition*, pour laquelle on utilise un module externe proposant un certain nombre d'étiquettes — ce qui revient à construire des contraintes de types ; et une étape de *désambiguïsation*, consistant à prédire en contexte la bonne étiquette parmi les propositions — ce qui revient à effectuer l'apprentissage en incluant les contraintes de

types à l'apprentissage. Pour les tâches considérées, il n'est pas envisageable de se passer des contraintes de types [11].

Müller *et al.* (2013) considèrent une autre manière de filtrer l'espace de recherche, dans le but d'utiliser un modèle CRF d'ordre plus important. Ces auteurs proposent ainsi d'utiliser une cascade de modèles de complexité croissante (Charniak et Johnson, 2005), en réduisant à chaque étape les étiquettes autorisées à chaque position.

Enfin, d'autres manières d'intégrer des contraintes dans un modèle ont été proposées. Dans la régularisation *a posteriori* (Ganchev *et al.*, 2010) la distribution est choisie de manière à maximiser la log-vraisemblance mais également à respecter, en moyenne, certaines contraintes. Un autre cadre permettant d'inclure des contraintes de manière déclarative est celui des modèles conditionnels sous contraintes (Chang *et al.*, 2010, 2012). D'autres manières d'intégrer l'information linguistique dans l'apprentissage supervisé et, plus généralement, de s'interroger sur la meilleure manière de choisir les compétiteurs lors de l'apprentissage, sont l'estimation contrastive (Smith et Eisner, 2005) et ses extensions récentes (Gimpel et Bansal, 2014).

4.5 Conclusions

Dans ce chapitre, nous avons exploré ce qui nous apparaissait comme un paradoxe, en essayant de répondre à la question suivante : pourquoi l'utilisation de contraintes utiles lors du décodage dégrade-t-elle les performances lorsque ces mêmes contraintes sont utilisées pour « aider » l'apprentissage ? Nous avons vu que l'intégration des contraintes lors de l'apprentissage conduit à ignorer la contribution de nombreux exemples, à savoir ceux qui seraient pleinement désambiguïsés par les contraintes, et que ceci nuit à la capacité de généraliser à des mots hors-vocabulaire.

De plus, il semble bien nécessaire de reproduire la même configuration à l'apprentissage et au test. Utiliser les contraintes extraites des données d'apprentissage conduit en réalité à une différence importante : l'absence de mots inconnus lors de l'apprentissage, le plus souvent entièrement désambiguïsés par les contraintes extraites, ce qui n'est plus le cas lors du test. La configuration pour laquelle on applique simultanément les contraintes à l'apprentissage et au test n'est donc pas, contrairement à ce que l'on peut penser, la situation théoriquement favorable où l'on reproduit les mêmes conditions.

Les contraintes de types, permettant d'accélérer considérablement l'apprentissage et le décodage des CRF, ne peuvent être utilisées telles quelles pendant l'ap-

11. Même avec celles-ci, Smith *et al.* (2005) font état de temps d'apprentissage de plusieurs jours.

prentissage sous peine de sévères dégradations des performances, même lorsque ces contraintes sont utilisées lors du décodage. Nous avons proposé quelques pistes permettant de limiter les effets négatifs tout en préservant les bénéfices en temps de calcul, ce qui est indispensable pour de nombreuses applications.

De manière plus générale, lorsque l'on dispose d'informations linguistiques, il semble important de faire attention à la manière dont on les intègre au modèle, car même une approche en apparence « inoffensive » peut se révéler néfaste pour les performances. La meilleure manière d'intégrer une information linguistique externe reste donc une problématique intéressante à étudier.

Deuxième partie

Le choix de l'espace des réordonnancements en traduction automatique

Chapitre 5

Le problème de la traduction automatique

Sommaire

Dans ce chapitre, nous présentons plus en détail la traduction automatique et le système état de l'art NCODE, point de départ de nos travaux sur le réordonnancement et que nous avons contribué à développer. Pour une présentation plus complète de la traduction automatique dans son ensemble et de ses évolutions récentes, le lecteur peut se référer à (Koehn, 2010) en anglais ou (Allauzen et Yvon, 2011) en français.

5.1 Le contexte de la traduction automatique

Depuis quelques années, la traduction automatique – un problème apparu très tôt en informatique – connaît un essort considérable, principalement en raison de l'avènement de l'Internet, tant par la disponibilité d'un nombre croissant de ressources que par les demandes massives et diverses de nombreux utilisateurs dans toutes les langues. Les méthodes statistiques, qui se basent sur le traitement de très grands corpus bilingues semblent les seules à pouvoir faire face à ces nouvelles exigences. Il subsiste malgré tout également une activité industrielle de traduction automatique à base de règles, focalisée sur la traduction d'énoncés répétitifs et contrôlés, dans des domaines de spécialité, comme par exemple la météorologique (Goldberg, 1993).

5.2 Une vue d'ensemble de la traduction statistique

Le problème de la traduction automatique est de construire un modèle capable, à partir d'une phrase *source* $\mathbf{x} = x_1 \dots x_I$, de produire une « bonne » traduction que l'on appelle phrase *cible* $\mathbf{y} = y_1 \dots y_J$.

Étant donné un modèle, le calcul de la traduction peut se formaliser comme un problème d'optimisation :

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) \tag{5.1}$$

où $s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})$ est une fonction de score pouvant prendre une certaine forme, paramétrée par un vecteur de paramètres $\boldsymbol{\theta} \in \mathbb{R}^d$.

Le processus de traduction étant très complexe, il est difficile de le modéliser directement, c'est-à-dire de définir $s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})$ simplement à partir de caractéristiques de $\mathbf{x}$ et de $\mathbf{y}$. On décompose alors le processus de traduction en étapes intermédiaires plus simples. L'ensemble des choix qui permettent de transformer une phrase source $\mathbf{x}$ en sa traduction $\mathbf{y}$ est appelé une *dérivation* notée $\mathbf{d}$.

L'équation (5.1) permet de mettre en évidence un certain nombre des défis et des difficultés du problème de la traduction automatique :

- Comment définir l'espace de recherche $\mathcal{Y}(\mathbf{x})$?
- Comment résoudre efficacement le problème de recherche du maximum ?
- Comment choisir la fonction de score ? Quelle est sa forme et sa structure ?
- Comment apprendre les paramètres de la fonction de score ?

5.2.1 Comment construire l'espace de recherche ?

En théorie, dans l'équation (5.1), on considère à partir d'une phrase source $\mathbf{x}$ l'ensemble de toutes les phrases de la langue cible et de toutes les dérivations possibles qui peuvent y conduire. En pratique, cet ensemble est infini, et toutes les approches de traduction automatique restreignent fortement l'espace de recherche $\mathcal{Y}(\mathbf{x}) \subset \mathcal{Y}$ des traductions possibles pour une phrase $\mathbf{x}$, ainsi que l'ensemble des dérivations possibles $D(\mathbf{x}) \subset D$ pouvant y mener. La forme précise de ces espaces dépend du paradigme de traduction choisi, mais on peut considérer qu'un système de traduction comporte deux modules : (a) un ensemble de contraintes qui permettent de construire l'espace de recherche exploré ; (b) un modèle utilisé pour évaluer les différentes hypothèses (Auli *et al.*, 2009).

5.2.2 Comment choisir la fonction de score ?

Comme souvent en TAL, on peut prendre un modèle linéaire $s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$. On peut également choisir un modèle probabiliste $s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$. Remarquons que ces deux cas sont équivalents si $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ est défini par un modèle exponentiel [1].

Les fonctions caractéristiques $\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$ (*features* en anglais) peuvent *a priori* être arbitraires et considérer n'importe quel aspect des phrases source et cible, comme des propriétés syntaxiques ou sémantiques globales. Par exemple, on peut avoir $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) = 1$ si la phrase y est « grammaticale » ou $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) = 1$ si $\mathbf{x}$ et $\mathbf{y}$ contiennent toutes les deux un verbe. Néanmoins, les modèles qui en résultent seraient pratiquement impossibles à entraîner et on est souvent contraint dans le choix des caractéristiques possibles que l'on peut ajouter au modèle. De telles caractéristiques globales apparaissent naturellement dans certains contextes et l'on souhaiterait pouvoir les intégrer dans les modèles, mais cela est délicat en raison de problèmes computationnelles. Par ailleurs, ces caractéristiques globales suffisent rarement à modéliser l'ensemble du processus, mais sont plutôt utilisées de manière complémentaire. Comme nous l'avons remarqué précédemment, il est difficile de construire directement des caractéristiques $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$ à partir de $\mathbf{x}$ et $\mathbf{y}$ en raison de la complexité du processus de traduction. On considère alors plutôt une fonction de score auxiliaire $s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{d})$ prenant en compte la dérivation $\mathbf{d}$ qui explicite le passage de $\mathbf{x}$ à $\mathbf{y}$.

En général la fonction de score $s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{d})$ est une fonction linéaire par rapport aux paramètres :

$$s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{d}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}, \mathbf{d}) \tag{5.2}$$

1. Inversement, remarquons que toute fonction de score peut être transformée en probabilité conditionnelle pour autant que la somme de l'exponentiel des scores converge.

où $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y}, \mathbf{d})$ est un vecteur de fonctions caractéristiques définies par rapport à une dérivation particulière. Pour des raisons computationnels, on suppose dans la plupart des cas que les caractéristiques se décomposent suivant la structure des dérivations. Nous verrons quelques exemples de caractéristiques possibles un peu plus loin.

Contrairement au problème de l'analyse morpho-syntaxique que nous avons rencontré aux chapitres 3 et 4, il n'apparaît pas de structure évidente suivant laquelle il est possible de décomposer le processus de traduction. Différentes approches ont ainsi été proposées : la traduction mot à mot, les modèles à base de segments (*phrase-based*), les modèles hiérarchiques ou les modèles syntaxiques. Dans l'approche à base de segments, que nous détaillons à la section 5.6, on peut considérer à quelques variantes près que le processus de traduction se décompose en trois étapes : (a) une étape de segmentation, qui consiste à découper la phrase source en unités formées de groupes de mots contigus ; (b) la permutation de ces segments pour rendre compte des divergences éventuelles de l'ordre des unités entre langue source et cible, étape que l'on appelle le *réordonnancement* ; et (c) la traduction de ces segments sources en segments cibles. Cette structure sous-jacente permet à la fois de définir l'espace de recherche et la forme que peuvent prendre les différents modèles et les caractéristiques qui les intègrent.

La figure 5.1 illustre ce processus : on choisit ① une segmentation $\bar{x}_1 \dots \bar{x}_K$ pour $\mathbf{x}$, ② une permutation σ induisant un réordonnancement de ces segments $\bar{x}_{\sigma(1)} \dots \bar{x}_{\sigma(K)}$, puis ③ on traduit chaque segment suivant une table de traduction $\bar{x}_{\sigma(k)} \rightarrow \bar{y}_k$ pour ④ construire la phrase cible $\mathbf{y} = \bar{y}_1 \dots \bar{y}_K$ par une désegmentation triviale. L'ensemble du processus qui permet de passer d'une phrase source à une phrase cible — composé de trois types de décisions : segmenter, permuter, traduire — constitue une dérivation. Remarquons qu'il peut bien sûr exister de nombreuses dérivations différentes d'une phrase $\mathbf{x}$ conduisant à une même traduction $\mathbf{y}$. Considérons la phrase « *It's a Russian Blue* » et une traduction possible en français : « C'est un bleu russe ». Une dérivation possible peut consister en trois segments « *It's* », « *a* » et « *Russian Blue* », traduits respectivement par « C'est », « un » et « bleu russe », sans réordonnancements. On pourrait également avoir un seul segment « *a Russian Blue* » traduit par « un bleu russe ». Ou encore deux segments « *Russian* » et « *Blue* », inversés puis traduits par « bleu » et « russe ». Toutes ces dérivations amènent à la même traduction. La dernière dérivation, si l'on omet l'inversion, engendrerait la traduction incorrecte [2] « C'est un russe bleu ».

Cependant, les dérivations ne sont pratiquement jamais annotées au niveau du corpus d'apprentissage. En traduction, on dispose de grandes quantités de corpus parallèles qu'il est possible d'aligner au niveau des phrases. Mais on ne dispose

2. Qui est pourtant une phrase française, mais avec un tout autre sens. On comprend sur cet exemple que la traduction automatique est une tâche ardue.

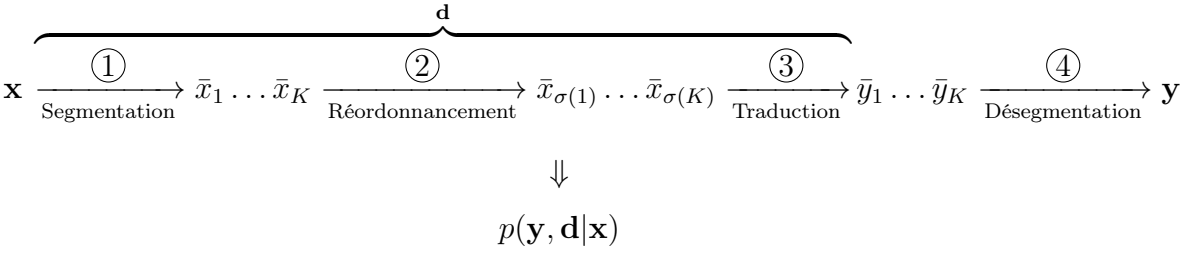


FIGURE 5.1 – Processus de traduction usuel.

pas de renseignements sur la manière dont les traductions ont étés obtenues. Une solution est alors de les considérer comme des variables latentes. L'équation (5.1) s'écrit alors

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \sum_{\mathbf{d}} s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{d}) \tag{5.3}$$

Cependant, marginaliser par rapport à $\mathbf{d}$ pour chaque valeur de $\mathbf{y}$ est un problème NP-dur en toute généralité et on a alors presque toujours recours à l'approximation consistant à remplacer la somme par le maximum

$$\mathbf{y}^* = \arg\max_{\mathbf{y}, \mathbf{d}} s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{d}) \tag{5.4}$$

Autrement dit, le choix de la fonction de score s'écrit donc soit :

$$s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = \max_{\mathbf{d} \in D(\mathbf{x})} s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{d}) \tag{5.5}$$

soit :

$$s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{d} \in D(\mathbf{x})} s_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{d}) \tag{5.6}$$

5.2.3 Architecture des systèmes de traduction

Depuis les travaux de (Och et Ney, 2002), on peut regrouper la plupart des approches en traduction statistique sous le formalisme précédent, un modèle exponentiel :

$$\mathbf{y}^* = \arg\max_{\mathbf{y},\mathbf{d}} p(\mathbf{y},\mathbf{d}|\mathbf{x}) \tag{5.7}$$

$$= \arg\max_{\mathbf{y},\mathbf{d}} \frac{\exp\left(\sum_{m=1}^{M} \lambda_m h_m(\mathbf{x},\mathbf{y},\mathbf{d})\right)}{\sum_{\mathbf{y}',\mathbf{d}'} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(\mathbf{x},\mathbf{y}',\mathbf{d}')\right)} \tag{5.8}$$

$$= \arg\max_{\mathbf{y},\mathbf{d}} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{x},\mathbf{y},\mathbf{d}) \tag{5.9}$$

Le changement de notations reflète le fait que les caractéristiques, ici notées h_m sont souvent plus complexes que celles que l'on trouve habituellement dans les modèles exponentiels en TAL et moins nombreuses [3]. Ces caractéristiques sont souvent elles-mêmes en réalité des sous-modèles, comme nous allons le voir. On peut aborder le problème précédent selon deux points de vue équivalents : une vision probabiliste (équation (5.7)) en reconnaissant un modèle exponentiel (équation (5.8)) ; ou une interprétation comme combinaison linéaire de modèles, calibrés par des poids λ_m (équation (5.9)).

En choisissant $M = 2$, $\lambda_1 = \lambda_2 = 1$, $h_1(\mathbf{x},\mathbf{y},\mathbf{d}) = \log p(\mathbf{y})$ et $h_2(\mathbf{x},\mathbf{y},\mathbf{d}) = \log p(\mathbf{x},\mathbf{d}|\mathbf{y})$ on retrouve le modèle classique du *canal bruité* qui consiste à appliquer la règle de Bayes à l'équation (5.7) pour décomposer cette probabilité en deux termes :

$$\mathbf{y}^* = \arg\max_{\mathbf{y},\mathbf{d}} p(\mathbf{y},\mathbf{d}|\mathbf{x}) = \arg\max_{\mathbf{y},\mathbf{d}} p(\mathbf{x},\mathbf{d}|\mathbf{y})p(\mathbf{y}) \tag{5.10}$$

Le modèle $p(\mathbf{y})$ est appelé *modèle de langue* et peut être construit sur un très grand corpus monolingue, le plus souvent suivant un modèle n-gram (Brown *et al.*, 1992). Le deuxième modèle peut encore se décomposer suivant :

$$p(\mathbf{x},\mathbf{d}|\mathbf{y}) = p(\mathbf{x}|\mathbf{d},\mathbf{y})p(\mathbf{d}|\mathbf{y})$$

où l'on voit que l'on choisit une dérivation possible, puis un choix de traduction connaissant cette dérivation. Ce cas particulier est généralisé par l'approche du modèle exponentiel, ce qui était la motivation essentielle de Och et Ney (2002). Remarquons que les deux caractéristiques ici sont elles-mêmes des sous-modèles qui peuvent prendre diverses formes [4].

Dans les premiers modèles, et même dans la plupart des modèles utilisés encore aujourd'hui, plusieurs caractéristiques du modèle exponentiel sont ainsi des

3. De l'ordre d'une dizaine pour les modèles de base.
4. Par exemple des modèles exponentiels !

sous-modèles : plusieurs modèles de langue, plusieurs modèles de traduction, des modèles de réordonnancement. Les poids λ_m de la combinaison linéaire — on dit aussi de *calibrage* — sont alors généralement optimisés sur un corpus de développement à l'aide d'une méthode discriminante, alors que les sous-modèles (les caractéristiques) sont appris préalablement sur de très gros corpus. Cela constitue la phase d'apprentissage qui se décompose ainsi en deux étapes :

1. Apprendre les différents paramètres des modèles génératifs qui constituent les caractéristiques du modèle exponentiel (h_m).
2. Calibrer les différents poids des modèles (λ_m).

Une fois les différents paramètres du modèle appris, on est en mesure de décoder une phrase inconnue $\mathbf{x}$. Le programme d'optimisation formulé par l'équation 5.7 requiert de comparer les probabilités $p(\mathbf{y}, \mathbf{d}|\mathbf{x})$ pour toutes les phrases cibles $\mathbf{y}$ que l'on peut obtenir à partir d'une dérivation $\mathbf{d}$, ce qui est bien sûr impossible étant donné la taille gigantesque de cet ensemble. On est alors obligé de contraindre l'espace de recherche d'une manière ou d'une autre et de recourir à des approximations.

5.2.4 Conclusion

On peut comprendre les différents systèmes de traduction par : (1) la manière dont sont modélisées les dérivations $\mathbf{d}$; (2) le choix des caractéristiques h_m ; (3) la manière d'optimiser les différents poids λ_m ; et (4) la manière de parcourir et de contraindre l'espace de recherche lors du décodage.

Dans la suite de ce chapitre, nous présentons rapidement le problème de l'alignement (§ 5.3) et celui du réordonnancement (§ 5.4), qui interviennent pratiquement toujours dans la manière de modéliser les dérivations et introduisons en quelques mots les problèmes d'évaluation (§ 5.5). Nous décrivons les modèles à base de segments (5.6), puis nous introduisons NCODE, un système à base de segments qui se distingue de l'approche usuelle sur différents points (§ 5.7). Il existe de nombreuses autres approches, dans lesquelles les dérivations peuvent par exemple correspondre à des séquences de réécritures de grammaires synchrones (Galley *et al.*, 2004; Chiang, 2005; Nesson *et al.*, 2006) qui ne suivent pas nécessairement le modèle proposé à la figure 5.1 mais que nous n'abordons pas ici.

5.3 Le problème de l'alignement

Le problème de l'alignement vise à chercher des correspondances mot à mot entre la phrase source et la phrase cible. Intuitivement, deux entités sont alignées si elles correspondent à des équivalents de traduction. Mais la notion de correspondance entre deux entités n'est pas nécessairement simple à définir en toute

généralité (Bellos, 2012). Pour une discussion plus complète sur le problème de l'alignement et une description des différents modèles, le lecteur peut se référer à (Tomeh, 2012). Par la suite nous ne nous intéressons qu'aux alignements entre mots [5]. On suppose ici que les phrases sont segmentées au niveau des mots, mais on pourrait envisager d'autres types de segmentation. Du fait de certaines expressions idiomatiques, de traductions non littérales ou de mots-outils qui n'existent que dans l'une des langues, l'alignement de mots peut-être une tâche délicate même pour un être humain. Comment aligner par exemple au niveau des mots la paire de phrase

> Attention à la marche en descendant de voiture
> Mind the gap between the train and the platform

qui peuvent pourtant être considérés comme des phrases en relation de traduction ?

Il existe de très nombreux modèles et méthodes pour traiter le problème de l'alignement, mais nous ne décrivons ici que brièvement les premiers modèles mot à mot appelés les modèles IBM (Brown *et al.*, 1993), développés dans le laboratoire de même nom au début des années 90.

Formellement, on définit un alignement entre deux phrases $\mathbf{x} = x_1 \dots x_I$ et $\mathbf{y} = y_1 \dots y_J$ par la donnée d'une matrice

$$\mathbf{A} \in M_{I,J}(\{0,1\})$$

Les mots x_i et y_j sont alors alignés si et seulement si $\mathbf{A}_{i,j} = 1$. Ce formalisme permet de considérer des alignements arbitrairement complexes. Cependant, on considère souvent un cadre plus restreint, où chaque mot cible y_j est aligné avec au plus un mot source, que l'on appelle un alignement *plusieurs-à-un*. Pour simplifier, on introduit un nouveau mot [6] $x_0 = \texttt{NULL}$ et on impose que chaque mot cible soit aligné avec exactement un mot source (ou avec le mot `NULL`). On définit alors formellement l'alignement comme une application

$$\begin{array}{rccc} \mathbf{a}: & J & \longrightarrow & I \\ & j & \longmapsto & a_j \end{array}$$

Intuitivement, on pourra penser que le mot x_{a_j} est le mot dont la présence en langue source justifie la présence de y_j en langue cible.

Les modèles décrits ici considèrent l'alignement $\mathbf{a}$ comme une variable cachée non observée dans le corpus d'apprentissage et qu'il faut donc traiter comme une variable latente :

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{a}} p(\mathbf{y}, \mathbf{a}|\mathbf{x}) \tag{5.11}$$

5. Ce qui impose de définir en premier ce qu'est un mot, ce qui n'est pas non plus évident (Bellos, 2012).

6. On introduit également $y_0 = \texttt{NULL}$ par commodité.

La probabilité conditionnelle se décompose comme :

$$p(\mathbf{y}, \mathbf{a}|\mathbf{x}) = p(J|I) \prod_{j=1}^{J} p(a_j|y_1 \dots y_{j-1}, a_1 \dots a_{j-1}, \mathbf{x}) p(y_j|y_1 \dots y_{j-1}, a_1 \dots a_j, \mathbf{x}) \tag{5.12}$$

en trois termes : une distribution sur la longueur des phrases, une probabilité d'alignement pour chaque position j et une probabilité lexicale de traduction [7] pour le mot x_j.

Les modèles IBM se distinguent ensuite en fonction des hypothèses d'indépendances conditionnelles faites sur ces distributions. Pour estimer les paramètres du modèle [8], du fait des variable latentes, la log-vraisemblance n'est pas convexe et on utilise l'algorithme *Expectation-Maximisation* (EM). Cet algorithme est sensible aux conditions initiales et on utilise alors une chaîne de modèles de complexité croissante, les résultats de l'un servant de conditions initiales au suivant.

Pour gérer le cas des mots non alignés, que l'on préfère considérer séparément pour de nombreuses raisons, on introduit un paramètre p_0 qui doit être optimisé sur un corpus de développement mais que l'on fixe en général à $p_0 = 0.99$. Chaque mot cible a une probabilité p_0 d'être aligné avec un mot source et une probabilité $p_1 = 1 - p_0$ de ne pas être aligné (i.e. d'être aligné avec le mot `NULL` en source).

Dans les trois modèles que nous présentons ci-dessous, le modèle de probabilité lexicale $p(y_j|y_1 \dots y_{j-1}, a_1 \dots, a_j, \mathbf{x})$ est simplifié en supprimant les dépendances sur les mots précédents $p(y_j|x_{a_j})$. Les différences portent donc sur les probabilités d'alignement.

IBM1 Le modèle IBM1 effectue l'hypothèse la plus forte et considère que tous les alignements sont équiprobables. La probabilité jointe d'une phrase cible et d'un alignement s'écrit alors :

$$p(\mathbf{y}, \mathbf{a}|\mathbf{x}) = \frac{p(J|I)}{(I+1)^J} \prod_{j=1}^{J} p(y_j|x_{a_j}) \tag{5.13}$$

L'avantage est que cela entraîne de nombreuses simplifications mathématiques : l'inférence peut être réalisée de manière exacte et l'on a une garantie de converger vers un maximum global dans EM. L'inconvénient est que l'on traite les phrases comme des sacs de mots et que les positions des mots ne jouent aucun rôle.

7. Ce modèle de probabilité lexicale peut lui-même être également utilisé dans l'équation (5.9) comme sous-modèle (i.e. caractéristique) (Koehn *et al.*, 2007).

8. Sauf pour le modèle IBM1 où il est possible de trouver une formule close.

IBM2 Le modèle IBM2 étend le modèle précédent en introduisant un modèle de distorsion $p(a_j|j, I, J)$.

$$p(\mathbf{y}, \mathbf{a}|\mathbf{x}) = p(J|I) \prod_{j=1}^{J} p(a_j|j, I, J) p(y_j|x_{a_j}) \tag{5.14}$$

Pour ce modèle, on a simplement la garantie de converger vers un maximum local et l'initialisation des paramètres devient critique : elle est typiquement réalisée en utilisant les paramètres donnés par IBM1.

HMM Ce modèle introduit par Vogel *et al.* (1996) propose plutôt de faire dépendre un lien d'alignement du lien précédent plutôt que de la position absolue dans la phrase. Le modèle de distorsion est alors $p(a_j|a_{prev}, I)$ où a_{prev} est le dernier lien d'alignement non aligné avec `NULL`.

$$p(\mathbf{y}, \mathbf{a}|\mathbf{x}) = p(J|I) \prod_{j=1}^{J} p(a_j|a_{prev}, I) p(y_j|x_{a_j}) \tag{5.15}$$

Ce modèle, qui permet une inférence exacte, est à la base de nombreuses extensions (Toutanova *et al.*, 2002; He, 2007).

Brown *et al.* (1993) introduisent ensuite des modèles de complexité supérieure — IBM3, IBM4 et IBM5 — conceptuellement assez différents en introduisant la notion importante de fertilité, qui permet de contrôler le nombre de mots cibles alignés avec un même mot source. Plutôt que de choisir une longueur pour la phrase cible puis d'aligner chacun de ses mots, on laisse chaque mot source engendrer un certain nombre de mots cibles, qui sont ensuite réordonnés.

Vu le formalisme choisi, les modèles précédents ne peuvent construire que des alignements *plusieurs-à-un* : plusieurs mots source peuvent être alignés à un même mot cible, mais l'inverse n'est pas vrai. Pour prendre ce phénomène en compte, on peut avoir recours à des modèles différents (Lardilleux et Lepage, 2009; Lardilleux *et al.*, 2013; Cromières, 2010) ou à des heuristiques de symétrisation (Och et Ney, 2003; Koehn, 2010).

5.4 Le problème du réordonnancement

Par réordonnancement, étape ③ de la figure 5.1, il faut comprendre la redistribution de l'ordre des mots de la phrase cible par rapport à la phrase source. Le réordonnancement peut porter sur les mots (comme dans les modèles d'alignement précédents) ou sur les unités de traduction (comme sur la figure 5.1). Le choix des unités de traduction implique d'ailleurs souvent une forme de réordonnancement

prenant en compte le contexte local et c'est peut-être là une des clefs du succès des méthodes à base de segments que nous présenterons plus loin. Cependant, ces réordonnancements locaux ne suffisent pas pour la majorité des paires de langues [9] et il est nécessaire de modéliser et de prendre en compte ce problème sérieusement. Nous reviendrons plus longuement sur le réordonnancement aux chapitres 6 et 7.

Costa-Jussà et Fonollosa (2009) proposent une classification des méthodes pour modéliser les réordonnancements qui nous semble assez pertinente :

Contraintes sur l'espace de recherche Comme l'espace de recherche engendré par tous les réordonnancements possibles est beaucoup trop large et que le problème du décodage dans cet espace est NP-dur, on peut imposer des contraintes sur celui-ci ou sur la manière de le parcourir. Parmi les contraintes typiques, mentionnons :

- Les contraintes IBM (Berger *et al.*, 1996) : à chaque étape, il faut choisir l'unité à traduire parmi les k premières non encore traduites.
- Les contraintes ITG (Wu, 1997) (*Inversion Transduction Grammar*) : les permutations autorisées sont celles qui peuvent être engendrées par une structure de branchement binaire où deux blocs adjacents peuvent être échangés.

Zens et Ney (2003) comparent l'utilisation des contraintes ITG et IBM. Zens *et al.* (2004) proposent un algorithme de programmation dynamique pour exploiter ces deux contraintes et Feng *et al.* (2010a) utilisent une approche *shift-reduce* pour les contraintes ITG en complexité linéaire.

Modèles de réordonnancement L'idée est d'avoir une ou plusieurs caractéristiques du modèle exponentiel qui pénalisent ou encouragent certains réordonnancements. Le modèle de réordonnancement le plus simple, le modèle de distorsion $dist(\bar{x}_k, \bar{x}_{k-1})$, pénalise le réordonnancement de ces deux unités, à quelques détails près, en fonction du nombre de mots entre les deux segments. Le modèle de réordonnancement lexical associe une probabilité d'orientation (qui peut être monotone, d'inversion ou discontinue) pour chaque unité bilingue. A la fois Moses et Ncode, utilisent ce modèle. Galley et Manning (2008a) étendent ce modèle au cas des dérivations hiérarchiques, ce qui permet de considérer les réordonnancements de plus longue distance.

Réordonnancement de la source L'idée est de réordonner préalablement la source dans un ordre plus proche de celui de la cible. Dans cette approche, on

9. Le problème du réordonnancement est d'ailleurs très sensible à la paire de langues considérée. Pourtant, la plupart des travaux utilisent les mêmes modèles de réordonnancement, quelle que soit la paire de langue considérée.

apprend, lors de la phase d'entraînement, à réordonner la source pour utiliser cela au décodage. On construit ensuite un graphe des réordonnancements possibles de la source à partir des règles apprises. C'est l'approche retenue par NCODE.

Réordonnancement syntaxique Des grammaires synchrones, des modèles hiérarchiques ou des traductions entre arbres syntaxiques sont utilisés pour permettre le réordonnancement de blocs et donc de longues distances. Le problème de ces modèles est que leur complexité reste généralement élevée. Cherry *et al.* (2012) étudient le lien entre cette approche et les contraintes sur l'espace de recherche.

Re-classement des N-meilleurs L'approche requiert ici d'utiliser de nombreuses informations syntaxiques pour reclasser les N-meilleurs candidats d'un système.

5.5 Le problème de l'évaluation

L'évaluation d'un système de traduction ou simplement déterminer ce qu'est une bonne traduction est un problème essentiel, complexe, subjectif, encore loin d'être résolu et suscitant de nombreux débats (Koehn, 2010). Smith (2012) propose d'ailleurs une autre approche pour l'évaluation en traitement automatique des langues. Les évaluations subjectives qui utilisent le jugement humain et qui semblent plus fiables et pertinentes [10], ont l'inconvénient d'être très coûteuses à réaliser, et non sans poser également de nombreux problèmes (Koehn, 2011). Pour pouvoir évaluer rapidement les systèmes lors de leur développement, on a alors recours à des métriques automatiques, imparfaites, mais que l'on espère corrélées avec ce que serait le jugement humain. La plupart de ces métriques, comme BLEU (Papineni *et al.*, 2002), TER (Snover *et al.*, 2006) ou METEOR (Banerjee et Lavie, 2005) se fondent sur l'idée qu'une bonne traduction est une traduction « proche » de celle que produirait un humain et comparent la sortie d'un système avec une ou plusieurs traductions dites de *référence*.

La métrique BLEU, pour *Billingual Evaluation Understudy*, se calcule en comparant le nombre de n-grammes communs entre la traduction candidate et celle d'au moins une des références. Pour une phrase $\mathbf{y}$ candidate, on définit la précision n-gramme $p_n(\mathbf{y})$ comme ce nombre divisé par le nombre de n-grammes dans $\mathbf{y}$. Le score BLEU est alors défini par :

10. Du moins plus proche de la réalité des applications : les traductions sont le plus souvent produites dans un but de lecture.

$$BLEU(\mathbf{y}) = \sqrt[4]{\prod_{n=1}^{4} p_n(\mathbf{y})} \tag{5.16}$$

ceci à quelques modifications près, entre autres l'ajout d'un terme *Brevity penalty* (BP) qui corrige le biais des candidats trop courts.

Candidat : `a compact car` `two` `, please .`

Référence 1 : `a two-door compact car` `, please .`

Référence 2 : `a compact car` `with` `two` `doors` `, please .`

FIGURE 5.2 – Calcul du score BLEU de la traduction proposée par NCODE de la phrase « une petite voiture à deux portes , s' il vous plaît . », $p_1(\mathbf{y}) = 7/7$, $p_2(\mathbf{y}) = 4/6$, $p_3(\mathbf{y}) = 2/5$, $p_4(\mathbf{y}) = 0/4$: BLEU $(\mathbf{y}) = 0$.

La figure 5.2 montre un exemple de calcul du score BLEU, à partir duquel on comprend que l'on ne peut pas calculer ce score sur des phrases isolées, mais qu'il importe de faire la moyenne arithmétique sur tout le corpus (du moins sur un grand nombre de phrases).

5.6 Les modèles à base de segments

L'approche la plus classique en traduction statistique, et qui constitue l'état de l'art de nos jours, est l'approche dite *à base de segments* dont MOSES [11] est l'implémentation de référence (Koehn *et al.*, 2007).

Dans cette approche, les unités de traduction sont des segments (*phrase* en anglais) de mots contigus. Pour définir ces segments, on utilise un corpus parallèle aligné au niveau des mots (§ 5.3) et l'on a recours à une heuristique pour extraire tous les bisegments $(\bar{x}_k, \bar{y}_k)$ compatibles avec les liens d'alignements. On peut se référer à Koehn (2010) pour plus de détails. On obtient alors un très gros dictionnaire de bisegments associés à des probabilités $\phi(\bar{x}_k|\bar{y}_k)$ et $\phi(\bar{y}_k|\bar{x}_k)$ estimées sur le corpus, qui permet à la fois de proposer une segmentation d'une phrase source (étape ① de la figure 5.1) et de fournir des estimations des probabilités de traduction segment par segment (étape ③).

En reprenant le modèle exponentiel introduit par l'équation 5.9, on peut voir le modèle standard comme le choix des caractéristiques :

11. http://www.statmt.org/moses/

$$h_1(\mathbf{y}, \mathbf{d}, \mathbf{x}) = \sum_k \log \phi(\bar{x}_{\sigma(k)}|\bar{y}_k) \qquad \text{(modèle de traduction)}$$

$$h_2(\mathbf{y}, \mathbf{d}, \mathbf{x}) = \sum_k \log dist(\bar{x}_{\sigma(k)}, \bar{x}_{\sigma(k-1)}) \qquad \text{(modèle de réordonnancement)}$$

$$h_3(\mathbf{y}, \mathbf{d}, \mathbf{x}) = \log p(\mathbf{y}) \qquad \text{(modèle de langue)}$$

On ajoute souvent à ce modèle diverses extensions, par exemple

$$h_4(\mathbf{y}, \mathbf{d}, \mathbf{x}) = \sum_k \log \phi(\bar{y}_k|\bar{x}_{\sigma(k)}) \qquad \text{(modèle de traduction inverse)}$$

$$h_5(\mathbf{y}, \mathbf{d}, \mathbf{x}) = J \qquad \text{(pénalité sur le nombre de mots)}$$

$$h_6(\mathbf{y}, \mathbf{d}, \mathbf{x}) = K \qquad \text{(pénalité sur le nombre de segments)}$$

Les différents paramètres du modèle exponentiel, régissant l'importance accordée à ces différents modèles, sont optimisés sur un corpus de développement. Il faut choisir un critère d'optimisation et l'on prend classiquement la métrique BLEU présentée ci-dessus en optimisant les paramètres à l'aide d'une méthode appelée MERT, pour *Minimum Error Rate Training* (Och, 2003). Il faut noter que cette étape est souvent de loin la plus coûteuse en terme de temps dans tout le processus d'apprentissage.

Lors du décodage, on parcourt implicitement les segmentations possibles (données par la table des segments) en exploitant les techniques de la section 5.4 pour contraindre les réordonnancements possibles. Cet espace gigantesque n'est pas exploré en totalité mais de nombreuses heuristiques et techniques d'élagage sont utilisées.

5.7 NCODE, une approche à partir de modèles de langages bilingues

Nous décrivons maintenant NCODE [12], un système de traduction open-source à base de n-grammes, qui a atteint des performances état de l'art en particulier dans les dernières campagnes d'évaluations internationales (Callison-Burch *et al.*, 2012; Bojar *et al.*, 2013, 2014). En parallèle à ce travail, nous avons pris part à son développement et aux dernières éditions des campagnes de traduction (Allauzen *et al.*, 2013; Wisniewski *et al.*, 2014; Pécheux *et al.*, 2014; Marie *et al.*, 2015; Wisniewski *et al.*, 2015). NCODE implémente l'approche n-gramme bilingue pour

12. http://ncode.limsi.fr

la traduction automatique (Casacuberta et Vidal, 2004; Mariño *et al.*, 2006; Crego et Mariño, 2006), similaire à celle à base de segments décrite précédemment (Zens *et al.*, 2002). Dans cette approche, le processus de traduction d'une phrase source $\mathbf{x}$ vers une phrase cible $\mathbf{y}$ est décomposé en deux étapes : une première étape de réordonnancement suivie d'une deuxième étape de traduction (monotone). La particularité de cette approche est d'utiliser des modèles n-grammes pour décomposer la probabilité d'une paire de phrases suivant une décomposition *bilingue* d'unités appelées *tuples*.

NCODE utilise un ensemble de caractéristiques combinées dans un modèle exponentiel de la même manière que l'approche standard à base de segments décrite précédemment (voir aussi Crego *et al.* (2011)). Dans le problème d'optimisation

$$\arg\max_{\mathbf{y},\mathbf{d}} p(\mathbf{y},\mathbf{d}|\mathbf{x}) = \arg\max_{\mathbf{y},\mathbf{d}} \frac{1}{\mathcal{Z}_{\mathbf{x}}} \exp\Big(\sum_{m=1}^{M} \lambda_m h_m(\mathbf{x},\mathbf{y},\mathbf{d})\Big) \tag{5.17}$$

la principale différence avec le modèle à base de segments est l'utilisation de caractéristiques h_m correspondant à des modèles de langues bilingues sur les tuples. Une dérivation $\mathbf{d}$ représente l'ensemble des dérivations cachées correspondant ici au réordonnancement et à la segmentation de la phrase source. En plus des modèles de traduction n-grammes qui font la particularité de cette approche, 13 autres caractéristiques standards sont combinées : 4 *modèles lexicaux* similaires à ceux utilisés dans l'approche à base de segments standard ; 6 *modèles de réordonnancement lexicalisés* (Tillmann, 2004; Crego *et al.*, 2011) qui visent à prédire l'orientation de l'unité de traduction suivante ; un simple *modèle de distorsion* ne prenant en compte que la distance ; et pour finir un *modèle de bonus de mot* et un *modèle de bonus de tuple* qui compensent la préférence du système à choisir des traductions courtes. Les caractéristiques sont estimées pendant la phase d'apprentissage et les poids associés (λ_m) sont estimés pendant la phase de calibrage sur des données de développement séparées. Les poids des différents modèles sont encore appris avec MERT comme dans le modèle à base de segments précédent.

L'idée sous-jacente est d'étendre le concept de modèle de langue à une langue constituée d'unités bilingues, elles-mêmes constituées de mots ou de propriétés sur les mots (Mariño *et al.*, 2006). Cela permet de bénéficier directement des nombreux travaux sur les modèles de langue. Pour cela, il est nécessaire de réordonner la source au préalable afin de pouvoir utiliser et estimer ce langage constitué de tuples bilingues. NCODE se distingue de l'approche précédente par le choix de ses unités de traduction, qui sont constituées de bisegments les plus petits possibles, appelés *tuples*. Le choix de prendre les unités les plus petites est motivé par le fait qu'elles seront amenées à être plus souvent réutilisées (elles sont moins rares). On ne considère donc plus vraiment le contexte local à l'intérieur des unités, mais la prise en compte de ce contexte est pour ainsi dire déléguée au modèle n-gramme de

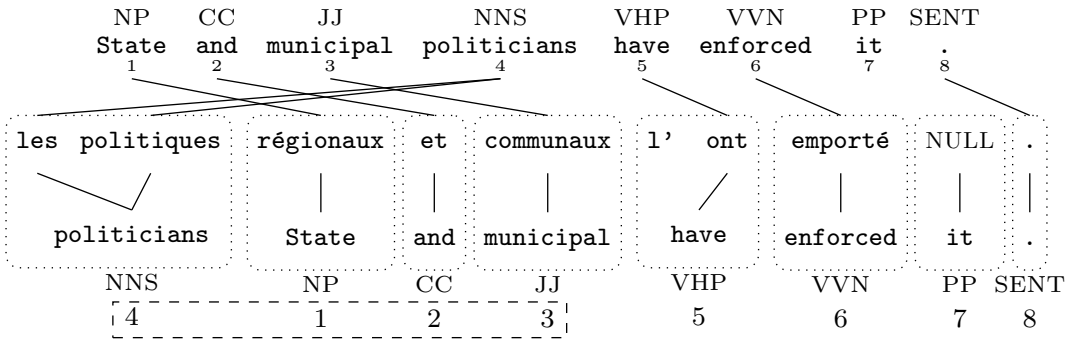


FIGURE 5.3 – Processus de démêlage et d'extraction des règles de réordonnancement à partir des alignements de mots. La phrase source en anglais est alignée avec sa traduction française. Le processus de démêlage implique un mouvement du quatrième mot « politicians » en tête de phrase, entraînant la permutation de la phrase source $\sigma = 41235678$. Les tuples qui seraient extraits sont entourés par une ligne en tirets. Le mot source non aligné 'it' est associé avec le token spécial `NULL`, le mot non aligné cible « l' » est attaché au tuple voisin. Une seule règle de réordonnancement serait extraite ici, associant à la séquence de PdD `NP CC JJ NNS` le réordonnancement minimal entouré sur la figure en pointillés.

langue bilingue. Pour choisir ces unités, on part toujours d'un alignement au niveau des mots comme précédemment, mais avant d'extraire les bisegments, on réordonne la source pour déplier (*unfolding*) ces liens suivant une heuristique qui permet d'extraire le plus de bisegments sans recouvrements possibles (Crego *et al.*, 2005). Cette procédure de dépliage, illustrée par la figure 5.3, permet une segmentation unique de la phrase source à partir de laquelle on peut extraire à la fois les tuples et des règles de réordonnancement. Par rapport à la figure 5.1, on peut considérer que les étapes ① et ② sont inversées : on commence d'abord par réordonner la source, puis on définit sa segmentation unique. Cette procédure de dépliage est extrêmement dépendante du bruit d'alignement qui affecte à la fois les tuples extraits et les règles de réordonnancement.

La procédure de dépliage est formalisée comme suit : la phrase source est tout d'abord segmentée en K segments de mots *consécutifs* $\mathbf{x} = \mathbf{x}_1 \dots \mathbf{x}_k \dots \mathbf{x}_K$ de manière à ce que pour chaque segment $\mathbf{x}_k$, si un mot cible y est aligné avec un mot de $\mathbf{x}_k$, alors tous les autres mots avec lesquels il est aligné sont aussi dans $\mathbf{x}_k$, c'est-à-dire si $\mathbf{y}_k = \{y \in \mathbf{y} \,|\, \exists x \in \mathbf{x}_k, (x, y) \in \mathbf{a}\}$ est l'ensemble des mots sources alignés avec $\mathbf{x}_k$, alors $\forall y \in \mathbf{y}_k, \forall x \in \mathbf{x}, (x, y) \in \mathbf{a} \Rightarrow x \in \mathbf{x}_k$. Cette procédure est la même que celle utilisée dans les modèles à base de segments standard pour

extraire les segments, à la différence que les mots de la phrase *source* ne sont pas ici nécessairement consécutifs. Il est ensuite possible de repositionner les mots sources réordonnés $\tilde{\mathbf{y}} = \mathbf{y}_1 \dots \mathbf{y}_k \dots \mathbf{y}_K$ (en utilisant l'ordre monotone au sein de chaque $\mathbf{y}_k$) [13] et de construire ainsi la séquence de tuples $\{(\mathbf{y}_k, \mathbf{x}_k)\}_k$. La figure 5.3 illustre cette procédure sur un exemple simple, où le mot `politicians` est déplacé au tout début de la phrase. Les mots cibles non alignés, comme par exemple « l' » à la figure 5.3, posent un problème étant donné que lors du décodage il est nécessaire d'avoir au moins un mot d'entrée pour engendrer une unité ; ils sont donc collés à l'un des mots voisins, celui qui maximise la probabilité lexicale (calculée par le modèle IBM 1 (de Gispert et Mariño, 2006)) du tuple ainsi formé. Les tuples sont ainsi extraits de manière à obtenir une unique segmentation du corpus bilingue. Un modèle de langage n-gramme est ensuite estimé sur le corpus d'apprentissage constitué de séquences de tuples.

Une fois la phrase source réordonnée et la segmentation en tuples bilingues effectuée, on peut apprendre différents modèles. En particulier des modèles de langue bilingues sur les tuples, soit sur les mots, soit sur les catégories grammaticales, ou encore un mélange des deux. On peut également apprendre un modèle de langue sur le côté source une fois celui-ci réordonné, ce qui permet d'évaluer la plausibilité de ces réordonnancements.

Lors du décodage, la phrase source est en premier lieu réordonnée de manière à reproduire au mieux l'ordre attendu de la langue cible. Cette étape aboutit à un treillis de réordonnancement constitué des permutations les plus vraisemblables (voir les détails au chapitre 6). Même restreint, l'espace ainsi engendré est trop grand pour permettre une recherche exacte, et NCODE utilise une méthode de recherche par faisceau faisant intervenir différentes piles. Comme l'estimation de coûts futurs est problématique en présence de multiples modèles n-grammes, NCODE utilise une pile pour chaque hypothèse recouvrant les *même mots sources*, par contraste avec une pile pour les hypothèses couvrant *le même nombre de mots* comme dans les modèles à base de segments standard. Ainsi, l'empreinte en mémoire de l'algorithme de décodage dépend directement du nombre de noeuds dans le treillis de réordonnancement.

Les réordonnancements sont appris suivant des règles de grammaire qui peuvent ensuite être généralisées (Crego et Mariño, 2006). Pour éviter le problème de parcimonie des données, ces règles sont plutôt apprises sur les catégories grammaticales. Ces règles de réordonnancement vont ensuite permettre d'engendrer un graphe des réordonnancements possibles et on considère ensuite pour une phrase source $\mathbf{x}$ toutes ses segmentations suivant la table des tuples, ce qui va définir l'espace de

13. Il est également nécessaire de choisir où placer les mots sources qui ne sont pas alignés, pour lesquels nous utilisons un tuple spécial avec un token spécial `NULL`(voir un exemple à la figure 5.3). Dans notre cas, ce mot est placé juste avant le prochain mot aligné.

recherche complet.

(Niehues *et al.*, 2011) intègrent également des modèles bilingues comme caractéristiques dans leurs modèles.

5.8 Calcul d'oracles dans un treillis

Dans ce travail, nous nous intéressons à l'espace de recherche d'un système de traduction. À ce titre, il est intéressant de savoir si l'espace considéré contient de bonnes hypothèses ou de connaître la qualité de la meilleure hypothèse du treillis encodant l'espace de recherche.

On s'intéresse donc à la possibilité de calculer les oracles, c'est-à-dire l'hypothèse dans un treillis ayant le meilleur score BLEU. Sokolov *et al.* (2014) décrivent différentes méthodes qui permettent de trouver les meilleurs oracles en BLEU sur un treillis. Pour que cette recherche soit efficace, il est nécessaire de disposer d'une approximation de BLEU (ou de toute autre métrique) qui se décompose sur les arcs du treillis. (Sokolov *et al.*, 2012) montrent que l'on peut utiliser l'approximation linéaire du score BLEU introduite par Tromble *et al.* (2008). Ces derniers montrent que l'approximation de Taylor au premier ordre du gain en BLEU au niveau du corpus (ou plutôt du log(BLEU)) conduit à une forme approchée du gain au niveau d'une phrase :

$$gain(\tilde{\mathbf{y}}, \mathbf{y}) = \theta_0 |\mathbf{y}| + \sum_{n=1}^{4} \sum_{g \in n\text{-gram}(\tilde{\mathbf{y}})} \theta_n \cdot \#_g(\mathbf{y}) \tag{5.18}$$

pour une référence $\tilde{\mathbf{y}}$, une hypothèse $\mathbf{y}$, avec n-gram($\tilde{\mathbf{y}}$) l'ensemble des n-grammes de $\tilde{\mathbf{y}}$ et $\#_g(\mathbf{y})$ désigne le nombre de fois qu'un n-gramme g apparaît dans $\mathbf{y}$. On peut en donner l'intuition suivante. Pour chaque mot d'une hypothèse donnée, ce mot a un gain par défaut de θ_0, puis, si ce mot apparaît dans la référence, on lui attribue également en plus un gain de θ_1, et ainsi de suite, c'est-à-dire que pour chaque n-gramme qui termine par ce mot, si ce n-gramme est dans la référence, on attribut à ce mot un gain θ_n. Au final, chaque mot de chaque chemin a un certain gain et l'on peut chercher le chemin de gain maximum par un simple algorithme de meilleur chemin. Cette approximation au niveau de la phrase se décompose comme une somme de fonctions locales, et il est donc possible de calculer de manière efficace le chemin possédant le gain maximum dans le treillis.

Les paramètres de l'équation (5.18) sont choisis en prenant comme bonus de longueur $\theta_0 = -1$ et pour $n \in \{1, \ldots, 4\}$ en utilisant la formule

$$\theta_n = \frac{1}{4p \times r^{n-1}} \tag{5.19}$$

où p désigne la précision unigramme et r le ratio de précision (Tromble *et al.*, 2008). La précision unigramme et le ratio de précision r sont choisis de manière à maximiser le score BLEU standard au niveau du corpus.

5.9 Conclusions

Au final, en dehors de quelques développements récents, l'architecture globale des systèmes de traduction est restée sensiblement stable depuis l'avènement des systèmes à base de segments et peut se décomposer en deux grandes étapes. Une première étape de collecte de statistiques, généralement effectuée sur de très grands corpus de données parallèles ou monolingues et une deuxième phase de calibrage des différents sous-modèles ainsi constitués, généralement coûteuse et effectuée sur de petits corpus du domaine d'intérêt.

Chapitre 6

Le problème des réordonnancements

Sommaire

Ce chapitre s'intéresse au problème des réordonnancements et plus précisément aux différentes approches qui ont été utilisées pour restreindre l'espace de recherche de ce point de vue. Nous présentons une étude détaillée d'un système à base de règles syntaxiques et le comparons à d'autres techniques usuelles. L'objectif est d'établir un diagnostic des performances du module de réordonnancement implanté dans NCODE et de sa capacité à reproduire les permutations de test. D'abord, dans ce chapitre, d'un point de vue intrinsèque, puis, dans le chapitre 7, du point de vue de l'application. Une partie de ce chapitre a été publiée dans (Pécheux *et al.*, 2014) et dans (Pécheux *et al.*, 2016a).

6.1 Le problème de l'ordre des mots

Le choix de l'ordre des mots dans une langue est un phénomène complexe et bien étudié en linguistique. En effet, le bon ordre est souvent un facteur constitutif de la langue et peut changer le sens d'une phrase. Pour certaines langues, l'ordre des constituants peut être relativement figé, c'est le cas par exemple en français, alors qu'il peut être beaucoup plus libre dans d'autres langues, comme le tchèque ou le latin. Le bon ordre des mots dans une phrase est donc un facteur important dans l'évaluation que peuvent en faire les lecteurs, même si certaines erreurs peuvent être moins graves que d'autres. Le problème de trouver le bon ordre des mots — et il en existe souvent plusieurs — est en soi une tâche de TAL qui a donné lieu à de nombreuses approches (de Gispert *et al.*, 2014). Elle n'est d'ailleurs pas inintéressante dans le contexte de la post-édition automatique, qui viserait par exemple à recouvrer le bon ordre dans des hypothèses produites par un système de traduction. Un autre problème intéressant serait, étant donné une phrase, de savoir circoncire la liste des changements d'ordre possible.

Les différences dans l'ordre des mots entre les langues source et cible est l'un des problèmes essentiels les plus complexes pour la traduction automatique. La complexité des réordonnancements pour une paire de langues donnée est d'ailleurs un assez bon indicateur de la difficulté à traduire automatiquement d'une langue à une autre (Birch *et al.*, 2008).

Pour traduire une phrase source, la plupart des systèmes de traduction automatique, que ce soit explicitement ou implicitement, sont obligés de choisir un ordre dans lequel lire la phrase source pour en engendrer des traductions partielles. Ceci induit une permutation, que l'on appelle un *réordonnancement*, des mots sources pendant le processus de traduction. Le problème du réordonnancement pose deux questions principales :

1. Quels sont les réordonnancements que l'on peut envisager ?
2. Comment choisir parmi ces réordonnancements ?

En premier lieu, l'exploration de l'espace de toutes les permutations, de taille factorielle, ne peut pas, sauf cas particulier, être envisagée. De manière générale, le problème de la traduction automatique est d'ailleurs NP-dur si l'on autorise toutes les permutations possibles (Knight, 1999). Même si cela était possible (par exemple pour des phrases courtes), cet espace contiendrait un grand nombre de permutations peu plausibles ou induisant de nombreuses ambiguïtés. Il est donc nécessaire, d'une manière ou d'une autre, de *réduire* l'espace des permutations que l'on accepte d'explorer, autrement dit d'imposer des *contraintes* sur les réordonnancements possibles. L'objectif est alors double : (a) l'espace de recherche ainsi constitué doit être assez grand pour donner lieu à un nombre suffisant de bonnes traductions ; (b) l'espace doit être le plus petit possible pour permettre un

décodage rapide et limiter les erreurs de recherche. Ce premier axe s'intéresse donc aux *contraintes de réordonnancement*.

Deuxièmement, il est nécessaire de trouver de bons modèles pour évaluer les candidats possibles parmi les réordonnancements considérés et pouvoir choisir la meilleure permutation possible. On peut utiliser par exemple des modèles simples qui prennent en compte la distance entre les mots réordonnés, des modèles de réordonnancement lexicalisés (Tillmann, 2004) ou encore des versions hiérarchiques de ceux-ci (Galley et Manning, 2008b), parmi de nombreuses autres possibilités. Ce deuxième axe s'intéresse ainsi à l'étude et à l'amélioration des *modèles de réordonnancement*. Notre approche a été de commencer par étudier le problème du réordonnancement suivant le premier axe avant d'aborder le deuxième, dans de futurs travaux. Nous verrons que les conclusions de l'étude que nous avons menée indiquent que le deuxième axe semble une perspective plus prometteuse.

Le problème des réordonnancements a été abordé sous de multiples angles depuis les débuts de la traduction automatique. Différentes approches originales ont été proposées pour attaquer ce problème, en adaptant de nouvelles stratégies de modélisation et en s'intéressant aux opérations possibles sur les changements d'ordre. Dans cette thèse, nous nous intéressons à la manière dont est défini l'*espace de réordonnancement*, qu'il soit explicite ou implicite, ainsi qu'à l'importance et aux conséquences de différents choix possibles.

Dans les approches à base de segments (§ 5.6), les réordonnancements de mots peuvent être divisés en deux phases qui restent fortement imbriquées : des réordonnancements locaux qui prennent place au sein même des segments ; et des réordonnancements de portée plus importante qui déplacent ensuite ces segments. De plus, une tendance émergente de ces dernières années consiste à utiliser comme prétraitement des méthodes de préordonnancement (§ 6.2) qui visent à réordonner la phrase source avant le processus de traduction lui-même, afin de lui donner un ordre plus proche de sa traduction de référence, et ainsi de simplifier la tâche du système de traduction en aval (Xia et McCord, 2004; Collins *et al.*, 2005; Tromble et Eisner, 2009; Genzel, 2010). Cette étape additionnelle complexifie grandement la compréhension des mouvements possibles des mots lorsque l'on s'intéresse au processus vu dans son ensemble. Enfin, en raison du nécessaire élagage de l'espace de recherche de la plupart des approches, l'espace des réordonnancements réellement considéré n'est en réalité qu'une sous-partie de celui défini formellement par les contraintes.

Dans ce chapitre, nous utilisons notre modèle de traduction état de l'art NCODE, décrit dans la section 5.7, qui permet de considérer séparément les étapes de réordonnancement et de traduction. L'ensemble des réordonnancements est encodé de manière compacte dans un treillis de permutations, *l'espace de réordonnancement*, qui sera ensuite traduit de manière monotone. Grâce à cette architecture, et

comme cet espace peut être construit par n'importe quelle méthode de réordonnancement [1], il est donc possible d'accéder directement à l'espace des permutations envisagées et ainsi de les étudier plus précisément.

6.2 Préordonnancement

Un des principaux problèmes qui se pose lorsque l'on cherche à intégrer les modèles ou les contraintes de réordonnancement directement dans le processus de décodage, est la difficulté à prendre en compte les réordonnancements faisant intervenir des déplacements importants, en raison de problèmes computationnels et de modélisation. Cela a motivé des approches de réordonnancement en amont du système de traduction qui essaient de transformer une phrase source de manière à lui conférer un ordre qui ressemble à celui de la langue cible visée. En général, ces approches ne sont pas limitées par la taille des réordonnancements à effectuer ni par les limitations de la modélisation des systèmes de traduction. Souvent le problème du réordonnancement est alors abordé directement au niveau des mots et non pas des segments.

Ce type d'approche a été initié par Xia et McCord (2004), qui, à partir d'arbres de dépendances sources et cibles, proposent d'apprendre automatiquement des règles de réordonnancement. Par la suite, d'autres travaux se sont attachés à construire des règles de réordonnancement à partir d'éléments syntaxiques ou d'arbres de dépendances ; soit manuellement (Collins *et al.*, 2005; Xu *et al.*, 2009; Carpuat *et al.*, 2010; Isozaki *et al.*, 2010b) ; soit en apprenant automatiquement les règles à partir des données (Xia et McCord, 2004; Zhang *et al.*, 2007; Li *et al.*, 2007; Khalilov *et al.*, 2009; Elming et Habash, 2009; Genzel, 2010; Dyer et Resnik, 2010; Khalilov et Sima'an, 2011; Lerner et Petrov, 2013). L'un des problèmes lorsque l'on utilise des arbres de dépendance vient du fait que ceux-ci ne sont pas toujours adaptés à la modélisation des déplacements que l'on cherche à retrouver. En effet, les réordonnancements de traduction ne suivent pas toujours les frontières définies par les arbres syntaxiques (Khalilov et Sima'an, 2012). De plus, des ressources annotées en arbres syntaxiques ne sont pas disponibles pour toutes les langues ou les domaines.

Une autre approche, dont fait partie celle que nous considérons dans ce travail, est d'apprendre des règles de réordonnancement dites « de surface », à partir d'annotations en catégories morpho-syntaxiques ou de segments syntaxiques ou sémantiques (Rottmann et Vogel, 2007; Zhang *et al.*, 2007; Crego et Habash, 2008;

1. À condition que l'espace ainsi défini puisse être encodé de manière compacte dans un treillis. Cependant, on pourrait étendre toute cette étude en incluant l'utilisation des hypergraphes, ce qui permettrait également de prendre en compte les méthodes produisant des permutations hiérarchiques.

Niehues et Kolss, 2009). Herrmann *et al.* (2013a) proposent également de combiner des règles de réordonnancement sur les PdD avec des contraintes structurelles dérivées d'arbres syntaxiques.

D'autres approches ont essayé de considérer le problème du réordonnancement directement sous la forme d'un problème de modélisation de permutations (Tromble et Eisner, 2009; Visweswariah *et al.*, 2011) ou encore d'induire automatiquement des « arbres de réordonnancement » de manière non-supervisée à partir de corpus parallèles (DeNero et Uszkoreit, 2011; Neubig *et al.*, 2012). Remarquons que ces approches sont cependant tributaires de la disponibilité d'alignements manuels, c'est-à-dire d'alignements d'excellente qualité. Comme on ne dispose de corpus annotés manuellement qu'en faible quantité, pour peu de domaines et pour un nombre de paires de langues restreint, Visweswariah *et al.* (2013) s'intéressent à une approche permettant d'améliorer simultanément la qualité des liens d'alignement et des réordonnancements lorsque l'on est en présence d'alignements automatiques et donc bruités.

Une dernière idée est de formuler le problème du réordonnancement comme un problème de traduction simplifié, en factorisant les mots par des classes de mots pour pouvoir généraliser. Ainsi on cherche à apprendre les mécanismes qui transforment une phrase source en sa contrepartie, dans la même langue, mais réordonnée (Costa-jussà et Fonollosa, 2006).

Ces techniques de préordonnancement ont abouti à des performances contrastées (Howlett et Dras, 2010). Howlett et Dras (2011) étudient en détail différents facteurs qui permettent de comprendre quand et à quelles conditions le préordonnancement peut être effectif. Zwarts et Dras (2006) suggèrent que la raison principale de leur succès est de permettre ensuite aux phrases ainsi réordonnées de mieux satisfaire les hypothèses des modèles à base de segments utilisés en aval.

Dans la plupart des approches ci-dessus, seul le meilleur réordonnancement trouvé par la technique de préordonnancement est ensuite passé à la chaîne de traitement de traduction. D'autres approches, comme la nôtre, préfèrent conserver à ce niveau un grand nombre de réordonnancements, et retarder le choix du meilleur réordonnancement en prenant en compte d'autres informations, lors du décodage. Les approches de préordonnancement sont en général simplement utilisées comme un premier prétraitement, qui est donc ensuite suivi de la chaîne standard d'un système de traduction, nouveaux réordonnancements compris. Par contraste, dans l'approche que nous suivons dans ce travail, tous les réordonnancements possibles sont directement définis par une même étape de réordonnancement, ce qui permet d'expliciter ainsi exactement l'espace de recherche de réordonnancement du processus global. Ceci permet de pouvoir comparer, de manière juste, n'importe quelle technique de préordonnancement ou de contraintes visant à engendrer l'espace des réordonnancements possibles.

6.3 Contraintes sur l'ordre des mots

De nombreuses contraintes sur les permutations possibles ont été proposées jusqu'alors, notamment IBM (Berger *et al.*, 1996), MJ (Kumar et Byrne, 2005) ou ITG (Wu, 1997). Ces contraintes ont été comparées en terme de performances (Zens et Ney, 2003; Zens *et al.*, 2004) ou à l'aide d'études oracles (Dreyer *et al.*, 2007; Wisniewski et Yvon, 2013). D'autres approches qui visent à donner à ces contraintes des fondements plus linguistiques se sont tournées vers des règles syntaxiquement informées, extraites automatiquement à partir des corpus d'apprentissage (Crego et Mariño, 2006; Niehues et Kolss, 2009; Herrmann *et al.*, 2013a). À notre connaissance, ces deux familles de contraintes, exclusivement combinatoires d'une part et issues de règles empiriques de l'autre, n'ont jamais été comparées de manière systématique.

Goh *et al.* (2011) proposent de segmenter une phrase source en différentes clauses et de restreindre les réordonnancements possibles à des réordonnancements à l'intérieur de ces clauses. De manière générale, la définition de l'espace des réordonnancements est par ailleurs intrinsèquement liée aux différents mécanismes sous-jacents des approches de traduction. Pour les modèles à base de segments (Zens *et al.*, 2002), les réordonnancements locaux sont modélisés au sein des segments, dont l'extraction peut obéir à différentes contraintes et heuristiques. Ces segment peuvent ensuite être réordonnés suivant certaines contraintes, par exemple une simple limite de distorsion. Plusieurs travaux ont proposé, pour déplacer les segments entre eux, d'utiliser les contraintes ITG (Zens *et al.*, 2004; Feng *et al.*, 2010b; Cherry *et al.*, 2012) ou MJ (Kumar et Byrne, 2005) [2].

Enfin, d'autres approches, comme les approches de traduction automatique à base de syntaxe, peuvent gérer le processus de réordonnancement de manière un peu différente, en suivant une analyse syntaxique pendant le processus de décodage (Wu, 1997; Yamada et Knight, 2001; Galley *et al.*, 2004), mais la compréhension précise de l'espace de réordonnancement implicite qui est exploré est alors encore plus délicate.

6.4 Les réordonnancements : une définition

Nous avons utilisé jusqu'ici le mot « réordonnancement » pour décrire le processus dans lequel l'ordre des mots de la phrase source se trouve changé pour produire celui de la phrase cible. Mais il n'est pas évident de saisir — et *a fortiori* de définir — ce que l'on entend par le « mouvement » des mots, étant donné que la traduction, de manière générale, ne se fait pas mot à mot, et que l'on ne retrouve

2. Nous utiliserons également les contraintes MJ, mais au niveau des *mots*, et non des *segments*.

donc pas nécessairement les mêmes mots dans la phrase à traduire et sa traduction. Nous avons vu au chapitre 5 que dans les modèles à base de segments, la phrase source est d'abord scindée en une suite de segments de mots, qui peuvent ensuite être réordonnés. Des « réordonnancements locaux » peuvent cependant apparaître au sein même de ces segments. Dans ce travail, nous nous intéressons au problème du réordonnancement dans son ensemble, c'est-à-dire en considérant l'intégralité des réordonnancements qui peuvent avoir lieu pendant le processus global, et il est donc naturel — bien que délicat — de chercher à se placer au niveau des mots, et donc des permutations de ces derniers.

Les réordonnancements de mots peuvent être, imparfaitement, inférés à partir des liens d'alignement, dont l'origine vient d'ailleurs des modèles de traduction mot à mot (Berger *et al.*, 1996). Il n'est cependant pas trivial d'induire une permutation à partir d'un alignement dans lequel plusieurs mots peuvent être alignés à un ou plusieurs mêmes mots, comme c'est le cas dans beaucoup d'alignements utilisés en traduction. De nombreuses heuristiques, qui peuvent différer par de nombreux détails ont été utilisées, par exemple pour l'évaluation du réordonnancement (Birch, 2011) ou pour développer des techniques de préordonnancement (Tromble et Eisner, 2009; Khalilov et Sima'an, 2012; Neubig *et al.*, 2012). Dans ce travail, nous utiliserons la technique de démêlage des liens d'alignement que nous avons rencontrée à la section 5.7 pour définir le réordonnancement. En effet, comme le rappelle la figure 5.3, démêler les liens d'alignement permet d'obtenir directement une permutation (au niveau des mots de la phrase cible), permutation que nous appellerons *réordonnancement de référence*.

Intuitivement, un réordonnancement se produit lorsqu'un mot « se déplace » par rapport à sa position d'origine. De manière générale, une permutation peut se décomposer en plusieurs réordonnancements locaux. Soit $\mathfrak{S}_n$ l'ensemble des permutations de $[\![1, n]\!]$ pour un entier n et soit $\sigma \in \mathfrak{S}_n$ une permutation $\sigma = \sigma_1 \ldots \sigma_n$. On définit un *réordonnancement* de σ comme étant une sous-séquence $\sigma_{[i:j]} = \sigma_i \ldots \sigma_j$ de σ avec $|j - i| > 1$ telle que

$$i \leq k \leq j \;\Rightarrow\; i \leq \sigma_k \leq j$$

i.e. $\{\sigma_k\}_{i \leq k \leq j} = \{i, \ldots, j\}$. Un réordonnancement est dit *minimal* si il est minimal pour cette propriété, c'est-à-dire s'il ne contient aucun autre réordonnancement autre que lui-même. La sous-séquence $\sigma_{[i:j]}$ correspond à la plus petite séquence (non triviale) qui doit être réordonnée pour retrouver la permutation identité. Il n'est pas difficile de voir qu'une permutation ne peut être ainsi segmentée que d'une seule manière, où chaque segment est soit un point fixe soit un réordonnancement minimal. Par exemple, le réordonnancement de référence $\sigma = 41235678$ de la figure 5.3 contient un unique réordonnancement minimal ($\sigma_{[1:4]} = 4123$) ainsi que quatre points fixes. À tout réordonnancement π on peut associer une (unique)

permutation $\bar{\pi} \in \mathfrak{S}_{|\pi|}$ quitte à réindexer les indices, i.e. $\forall k, \bar{\pi}_k = \pi_k - \min(\pi)$. Soit $\mathcal{R}_n \in \mathfrak{S}_n$ l'ensemble des réordonnancements minimaux de $[\![1, n]\!]$ pour $n \geq 2$. Le nombre de réordonnancements minimaux $r_n = |\mathcal{R}_n|$, peut être calculé de manière récursive par la formule suivante [3]

$$r(n) = \begin{cases} 1 & si \quad n = 1 \\ n! - \sum_{i=1}^{n-1} r(i) \cdot (n-i)! & sinon \end{cases} \tag{6.1}$$

où l'on a pris $r(1) = 1$ par commodité mathématique.

6.5 Des règles pour limiter les réordonnancements

Dans ce chapitre, nous étudions en détail le système à base de règles de NCODE. Dans ce système, les règles de réordonnancement sont extraites pendant la phase d'apprentissage, directement à partir des données d'entraînement (§ 6.5.1). Pour mieux généraliser à des configurations non observées, on utilise des facteurs de mots (syntaxiques ou non) plutôt que les mots-formes directement. Nous étudions en détail la capacité de ces règles à définir un espace de recherche précis et complet. Nous montrerons dans le chapitre 7 que l'utilisation de ces règles linguistiquement informées permet de définir des espaces de réordonnancement de taille réduite tout en permettant une traduction de meilleure qualité. Enfin, dans ce chapitre et le suivant, nous montrons que le choix du jeu de facteurs, qui influe sur la capacité des règles à bien généraliser, n'a au final que peu d'impact sur les performances globales.

6.5.1 Extraction des règles de réordonnancement

Les règles de réordonnancement sont automatiquement extraites lors du processus de démêlage des liens d'alignement qui aboutit à la segmentation en tuples (§ 5.7). Soit $\mathbf{x} = x_1 x_2 \ldots x_n$ une phrase source et $\mathbf{t} = t_1 t_2 \ldots t_n$ la séquence d'étiquettes associée. Soit $\mathbf{x}_\sigma = x_{\sigma_1} x_{\sigma_2 \ldots} x_{\sigma_n}$ la séquence réordonnée obtenue par démêlage, avec $\sigma = \sigma_1 \ldots \sigma_n \in \mathfrak{S}_n$. Une règle est extraite pour chaque réordonnancement minimal $\sigma_{[i:j]}$ de σ. Les règles ont ainsi la forme suivante :

$$\mathbf{t}_{[i:j]} \rightarrow \bar{\sigma}_{[i:j]}$$

3. On obtient ce résultat en étendant la définition pour autoriser les réordonnancements à être de taille 1 (c'est-à-dire que les points fixes sont également des réordonnancements minimaux). On compte ensuite le nombre de réordonnancements qui ne sont pas minimaux, ensemble que l'on écrit $\bar{\mathcal{R}}_n$ dont le cardinal est noté $\bar{r}(n)$. Soit $\sigma \in \bar{\mathcal{R}}_n$, et soit σ_1 le plus petit réordonnancement de σ qui en soit aussi un préfixe. Il est nécessairement non trivial, donc $\sigma = \sigma_1 \sigma_2$ avec $\sigma_1 \in \mathcal{R}_i$ et $\sigma_2 \in \mathfrak{S}_{n-i}$. Cette décomposition est unique et décrit exactement les permutations dans $\bar{\mathcal{R}}_n$, ceci définit donc une bijection entre $\bar{\mathcal{R}}_n$ et l'union disjointe $\cup_{1 \leq i \leq n-1} \mathcal{R}_i \times \mathfrak{S}_{n-i}$, ce qui nous donne la formule proposée.

où $\bar{\sigma}_{[i:j]}$ est la permutation induite de $\mathfrak{S}_{|j-i+1|}$ obtenue en réindexant $\sigma_{[i:j]}$ comme décrit à la section 6.4. La longueur de la règle est la longueur $j - i + 1$ de la permutation induite. On trouve un exemple sur la figure 5.3 où seule la règle

$$\text{NP CC JJ NNS} \rightarrow 4\ 1\ 2\ 3$$

serait extraite.

Il serait également possible d'extraire toutes les règles $\mathbf{t}_{[i:j]} \rightarrow \bar{\sigma}_{[i:j]}$ pour n'importe quel intervalle $|j-i| > 1$, mais des expériences préliminaires ont montré une légère baisse de performances pour cette variante, qui n'est pas davantage explorée dans ce travail.

On peut pondérer les règles avec un coût qui dépend de leur fréquence :

$$\text{coût}(\mathbf{t} \rightarrow \sigma) = -\log \frac{\#(\mathbf{t} \rightarrow \sigma)}{\sum_{\sigma' \in \mathfrak{S}_{|\mathbf{t}|}} \#(\mathbf{t} \rightarrow \sigma')}$$

où $\#()$ désigne le compte calculé sur le corpus d'apprentissage. Afin de réduire le bruit dû aux erreurs d'alignement et de limiter la taille de l'espace de réordonnancement, les règles trop rares sont filtrées si leur coût dépasse un certain seuil. Étant donné que le coût est l'opposé du logarithme d'un ratio conditionnel, un jeu d'étiquettes plus général peut entraîner un élagage plus sévère qu'un jeu d'étiquettes à grain fin, et ainsi conduire à extraire un ensemble de règles plus réduit. Le seuil optimal dépend donc de la granularité du jeu d'étiquettes utilisé pour abstraire les mots-formes ainsi que de la paire de langues et de la direction de traduction.

Les règles peuvent également être limitées en fonction de leur longueur. Des expériences préliminaires — et nous en verrons les raisons par la suite — montrent qu'augmenter cette taille maximale ne change en rien les performances. En fait, les règles longues sont le plus souvent trop rares pour se généraliser au delà de l'ensemble d'apprentissage duquel elles sont extraites. Les réordonnancements de longue distance, c'est-à-dire ceux dont la longueur excède la longueur de règle maximale, sont ainsi explicitement exclus du modèle. Remarquons que c'est également le cas dans les modèles à base de segments standard, où la taille maximale des réordonnancements, directement donnée par la limite de distorsion, est même le plus souvent limitée à une valeur encore plus faible. Cette limite de distorsion est par exemple égale à 6 par défaut dans Moses.

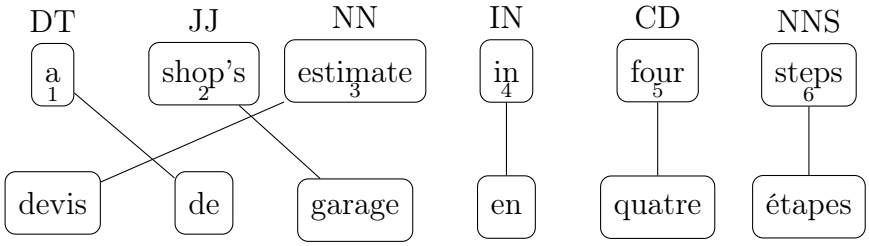
6.5.2 Jeux d'étiquettes

Dans (Crego et Mariño, 2006), les règles de réordonnancement sont fondées sur les parties du discours, plutôt que sur les formes de surface afin d'augmenter la puissance de généralisation. Cependant, il est tout à fait possible d'utiliser n'importe quelle autre abstraction possible des mots ou de considérer des PdD de

différentes granularités. Dans le but de mieux comprendre l'importance des différents niveaux de généralisation ainsi que la pertinence des facteurs syntaxiques, nous introduisons différents jeux d'étiquettes possibles, en plus des étiquettes de parties du discours (PdD) :

- **Une seule étiquette** : ce jeu d'étiquettes (virtuel) ne comporte qu'une seule étiquette, c'est-à-dire que l'on considère ici que tous les mots sont équivalents. Les règles de réordonnancement sont donc extraites et appliquées indépendamment de toute information contextuelle ou syntaxique. Ceci aboutit à un système qui autorise toutes les permutations [4] qui sont observées dans les corpus d'apprentissage.
- **Parties du discours universelles** (PDU) : ce jeu d'étiquettes est réduit aux 12 étiquettes de parties du discours universelles, simplifiées, ce qui introduit une dépendance vis-à-vis de la syntaxe, mais limite le nombre de règles par rapport à un jeu d'étiquettes plus détaillé. Comme pour les chapitres 3 et 4, nous utilisons le jeu d'étiquettes universelles proposé par (Petrov *et al.*, 2012b). Pour des langues peu dotées en ressources, ces étiquettes universelles peuvent être projetées par transfert cross-lingue ou apprises à partir d'annotations partielles (§ 3), ce qui permet ainsi de pallier l'éventuelle absence d'outils pour une langue pour laquelle on ne disposerait pas d'analyseur morpho-syntaxique.
- **PdD partiellement lexicalisées** (PdD|50) : pour ce jeu d'étiquettes, les étiquettes des 50 mots les plus fréquents sont lexicalisées, c'est-à-dire se voient attribuées une étiquette spécifique, les autres mots étant étiquetés par des PdD usuels. Les règles ainsi extraites, à rapprocher des règles lexicalisées de (Huang et Pendus, 2013), sont plus spécifiques qu'avec le jeu de PdD standard.
- **Classes de Brown** (classe) : les classes de mots, apprises de manière non supervisée, sont une première approximation des étiquettes de parties du discours lorsqu'un analyseur syntaxique n'est pas disponible. Dans (Ramanathan et Visweswariah, 2012), les classes de mots conduisent à des résultats légèrement dégradés, mais tout à fait raisonnables, par rapport aux PdD, dans une tâche de préordonnancement. Dans ce travail, nous utilisons les classes de mots données par les méthodes statistiques de Brown *et al.* (1992).
- **Mots-formes** (mot) : enfin, nous utilisons les mots pour construire des règles entièrement lexicalisées, ce qui produit des règles très spécifiques, mais dont on peut douter de la généralisation hors des corpus d'apprentissage.

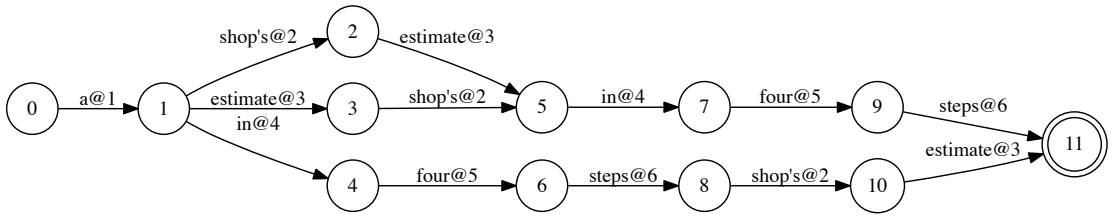
4. Au filtrage près des permutations rares.



(a)

JJ NN → 1 0 [coût=1.16]
JJ NN IN CD NNS → 2 3 4 0 1 [coût=3.90]

(b)



(c)

FIGURE 6.1 – Exemple de phrase source annotée, avec la phrase cible correspondante et l'alignement forcé entre les deux (a) ; les règles de réordonnancement qui s'appliquent pour cette phrase source (b) ; le treillis de réordonnancement que l'on obtient lorsque l'on applique ces règles (c).

6.6 Génération de l'espace des réordonnancements

Un *treillis de permutations* (Crego, 2008) est un automate fini $\mathcal{L} = \langle V, E, \Sigma, w \rangle$, où V désigne un ensemble d'états, E l'ensemble des transitions, l'alphabet $\Sigma = \{1, \ldots, n\}$, $w : E \to \mathbb{R}$ la fonction de poids ; qui engendre (ou reconnaît) un langage $L(\mathcal{L}) \subset \mathfrak{S}_n$, c'est-à-dire que chaque chemin dans l'automate correspond exactement à une permutation de $\{1, \ldots, n\}$. Pour certains sous-ensembles de permutations, un treillis permet d'encoder un nombre exponentiel de permutations avec un nombre polynomial d'états (et de transitions). Une propriété importante d'un treillis de permutations, qui nous sera utile par la suite est que l'ensemble des chemins qui arrivent sur un état donné contient exactement les mêmes indices, dans un ordre différent.

Un treillis de permutations est construit à partir d'une phrase source $\mathbf{x}$ associée à sa séquence d'étiquettes $\mathbf{t}$ comme suit. Un chemin monotone contenant

la permutation *identité* est construit en premier. Ensuite, pour chaque segment $[i : j]$ et chaque règle de réordonnancement $\mathbf{t}_{[i:j]} \rightarrow \sigma$, le treillis est étendu en ajoutant le sous-chemin $\sigma([i : j])$. La figure 6.1 illustre ce procédé par un exemple. Les règles sont appliquées en parallèle et n'interagissent pas entre elles. La prise en compte de toutes les règles de réordonnancement aboutit ainsi à un graphe fini, le treillis de permutations, qui représente l'espace de réordonnancement. Ce treillis peut être pondéré, par exemple en utilisant les probabilités des différentes règles, comme c'est le cas dans (Herrmann *et al.*, 2013b). Le score d'un chemin du treillis peut alors être utilisé dans la combinaison linéaire, comme une caractéristique supplémentaire — et donc comme un modèle de réordonnancement— dans l'équation (5.7), mais nous n'avons pas suivi cette possibilité qui n'a pas permis d'obtenir de meilleures performances dans nos expériences préliminaires.

En principe, il est possible de considérer n'importe quel sous-ensemble de permutations et de l'encoder dans un treillis. En pratique, comme nous l'avons vu à la section 5.7, le nombre d'états du treillis doit rester raisonnable (polynomial) par rapport au nombre de mots de la phrase source [5]. Afin de pouvoir mesurer l'intérêt de contraindre les réordonnancements à ceux uniquement observés sur le corpus d'apprentissage et la pertinence de l'utilisation de règles syntaxiques, nous comparons cette approche avec le sous-ensemble de permutations donné par les contraintes MJ (Kumar et Byrne, 2005). Dans les contraintes MJ-i, un mouvement de mot ne peut excéder i positions. L'appellation Max-Jump (MJ) vient ainsi du fait qu'un mot ne peut pas « faire un saut » de plus de i. Remarquons que puisque l'on considère ces contraintes sur les mots, cela revient également à considérer toutes les permutations possibles sous la contrainte de distorsion inférieure à $i + 1$. C'est également équivalent à un système à base de règles contenant toutes les règles possibles, sous la contrainte que leur taille soit plus petite que $i + 1$.

6.7 Mesures de complexité des réordonnancements

Nous avons choisi de décrire la complexité des réordonnancements en fonction de leur taille par trois indicateurs : la proportion de règles qui sont ITG ; le τ Kendall moyen (normalisé) ; et la distance moyenne (normalisée) de fragmentation entre un réordonnancement et la permutation *identité* de la taille de ce réordonnancement.

5. Ce ne serait pas le cas pour les contraintes ITG par exemple, que l'on ne peut pas encoder de manière compacte dans un treillis du fait de leur nature globale. Cependant, encore une fois, il serait possible sans grands changements d'étendre nos travaux aux systèmes hiérarchiques et de considérer alors des *forêts de permutations* au lieu de simples treillis, afin de pourvoir prendre en compte des espaces de réordonnancement plus généraux (Dyer et Resnik, 2010).

La famille des permutations ITG est un sous-ensemble de permutations « simples », et le nombre de réordonnancements non-ITG peut ainsi être utilisé comme une mesure de complexité des réordonnancements mis en jeu. Pour les deux autres indicateurs nous utilisons deux métriques sur les permutations. Le τ de Kendall (Kendall, 1962) compte le nombre d'inversions entre deux permutations $\sigma, \pi \in \mathfrak{S}_n$

$$\tau(\sigma, \pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbb{1}_{\{\sigma_i < \sigma_j\}} \mathbb{1}_{\{\pi_i > \pi_j\}} \tag{6.2}$$

où $\mathbb{1}_{cond}$ est la fonction indicatrice valant 1 si la condition *cond* est vraie et 0 sinon. C'est également le nombre minimal de transpositions entre symboles adjacents nécessaire à la transformation d'une permutation en une autre, ce qui explique que cette distance est également parfois appelée la distance du *tri à bulle*. Le τ est généralement normalisé, de manière à ce que la valeur 1 indique un désaccord maximal :

$$\tau^{\text{norm}} = \sqrt{\frac{2\tau(\sigma, \pi)}{n(n-1)}} \tag{6.3}$$

La distance de fragmentation, utilisée notamment pour évaluer le réordonnancement dans METEOR (Banerjee et Lavie, 2005), consiste à compter le nombre de sous-séquences communes entre deux permutations, c'est-à-dire à évaluer en combien de segments il est nécessaire de partitionner une permutation pour obtenir l'autre en réordonnant ces segments. La distance de fragmentation est alors ce nombre de segments moins un. Remarquons que c'est également le nombre de « sauts » qu'il est nécessaire d'effectuer pour lire une des deux permutations afin de retrouver l'autre, ce qui intuitivement semble proche de la complexité de réordonnancement pour un être humain et qui a motivé son utilisation (Talbot *et al.*, 2011). Enfin, cette distance correspond également au nombre de bi-grammes qui diffèrent lorsque les deux permutations sont vues comme des séquences de mots. Formellement, la distance de fragmentation entre deux permutations $\sigma, \pi \in \mathfrak{S}_n$, pour $n \geq 2$ est donc :

$$frag(\sigma, \pi) = \sum_{i=0}^{n-2} \mathbb{1}_{\{\pi^{-1}(\sigma_i)+1 \neq \pi^{-1}(\sigma_{i+1})\}} \tag{6.4}$$

Les extrémités jouent cependant dans cette formule un rôle moins important, en n'apparaissant que dans un seul bi-gramme contre deux pour les positions intermédiaires. Comme dans (Neubig *et al.*, 2012), nous ajoutons artificiellement une position initiale et finale dans le calcul de l'équation 6.4 afin de prendre en compte des mouvements du premier et/ou du dernier indice. Comme pour le τ de Kendall, on utilise une version normalisée :

$$frag^{\text{norm}}(\sigma, \pi) = \frac{frag(\sigma, \pi)}{n - 1} \tag{6.5}$$

Les statistiques sont calculées au niveau des réordonnancements, et donc des règles, plutôt qu'au niveau des phrases, étant donné que les phrases peuvent se décomposer en plusieurs segments qui peuvent être indépendamment réordonnés. Pour les phrases longues avec de nombreux réordonnancements locaux indépendants, les propriétés de chaque réordonnancement sont plus pertinentes que celles de la phrase considérée comme un tout. Par exemple, une phrase non-ITG, peut cependant être composée de nombreux réordonnancements ITG mais n'avoir qu'un seul réordonnancement non-ITG.

6.8 Résultats expérimentaux

6.8.1 Cadre expérimental

Notre cadre expérimental est basé sur la campagne d'évaluation internationale WMT [6] qui propose plusieurs tâches d'évaluation, des corpus et des outils génériques. Nous avons choisi trois paires de langues : anglais-français, anglais-allemand et anglais-tchèque. Comme corpus d'apprentissage nous utilisons NEWS-COMMENTARY dans sa version donnée par les organisateurs de WMT'12 (Callison-Burch *et al.*, 2012), et utilisons NEWSTEST2009 et NEWSTEST2010 respectivement comme corpus de développement et comme corpus de test [7]. Le tableau 6.1 contient quelques statistiques relatives à ces corpus.

Nous utilisons des outils internes pour la segmentation en unités linguistiques (Déchelotte *et al.*, 2008) et ne conservons l'information de casse que lorsque celle-ci est utile (Allauzen *et al.*, 2013). L'allemand est une langue à morphologie riche et complexe. Par conséquent, lorsqu'elle est utilisée comme langue source, elle est de plus normalisée en utilisant un prétraitement particulier (Allauzen *et al.*, 2010; Durgar El-Kahlout et Yvon, 2010) qui vise à réduire les redondances lexicales en normalisant l'orthographe, en supprimant les nombreuses flexions propres à cette langue et en scindant les mots-formes complexes. Pour annoter le côté anglais des corpus parallèles avec les parties du discours, nous utilisons Wapiti [8] (Lavergne *et al.*, 2010b) pour le français, TreeTagger (Schmid, 1994) pour l'allemand et enfin le système MORPHODITA [9](Straková *et al.*, 2014) pour le thèque. Pour cette

6. Workshop on Machine Translation — voir `www.statmt.org/wmt`.
7. Remarquons que la partie anglaise de NEWSTEST2009 et de NEWSTEST2010 est la même pour toutes les directions de traduction, ce qui permet une comparaison juste par la suite.
8. Avec le modèle par défaut disponible à l'adresse `https://wapiti.limsi.fr`.
9. Avec les modèles de Straka et Straková (2013). Voir http ://ufal.mff.cuni.cz/morphodita.

	NewsCommentary					Newstest2010					# PdD
	phrase		% mono	reord.		phrase		% mono	reord.		
	#	long.		#	long.	#	long.		#	long.	
$en \rightarrow fr$	137k	25	17	1.8	3.8	2k	25	20	1.6	4.3	44
$fr \rightarrow en$		29	14	2.0	4.7		28	17	1.7	4.9	34
$en \rightarrow de$	158k	24	19	1.6	6.4	2k	25	17	1.5	7.2	44
$de \rightarrow en$		26	16	1.7	6.8		26	16	1.6	7.3	116
$en \rightarrow cs$	139k	23	31	1.0	6.5	2k	25	27	1.1	6.9	44
$cs \rightarrow en$		21	29	1.0	6.3		21	29	1.0	6.6	63

Tableau 6.1 – Quelques statistiques sur les corpus utilisés : nombre (#) et taille moyenne (long.) des phrases sources (phrase) ; pourcentage de paires de phrases monotones (mono) ; nombre moyen par phrase (#) et taille moyenne (long.) des réordonnancements (reord.) ; ainsi que la taille du jeu d'étiquettes de PdD de la langue source.

dernière langue, nous n'utilisons que les deux premiers caractères, parmi les 15, des étiquettes du corpus arboré *Prague Dependency Treebank*, qui représentent les 63 PdD pour le tchèque. Pour toutes les langues, nous avons adapté les tableaux associatifs de Petrov *et al.* (2012b) pour simplifier les PdD en étiquettes universelles.

Les alignements au niveau des mots et les 50 classes de mots [10] sont inférés en utilisant respectivement mgiza++ [11] et mkcls [12] avec les paramètres par défaut, en utilisant, pour l'anglais-français et pour l'anglais-allemand, tous les corpus parallèles décrits dans (Allauzen *et al.*, 2013) et, pour l'anglais-tchèque, les corpus Europarl et CommonCrawl de l'édition WMT'12.

Pour chaque tâche, un modèle de langage 4-grammes est estimé en utilisant simplement le côté cible des données d'apprentissage (NewsCommentary). Nous utilisons Ncode avec les paramètres par défaut, en ajoutant un modèle additionnel basé sur les facteurs de mots, ici les PdD [13]. Le nombre maximal d'hypothèses par noeud dans la recherche en faisceau est fixé à 25 pendant la phase de développement et à 50 lors du décodage, cette différence étant bénéfique selon des résultats d'ex-

10. Les mots hors-vocabulaire lors du décodage sont associés à la classe 1.
11. http://www.kyloo.net/software/doku.php
12. http://code.google.com/p/giza-pp/
13. Remarquons que ceci est entièrement indépendant du choix du jeu d'étiquettes utilisé par les règles de réordonnancement.

périences préliminaires. Tous les chiffres de résultats sont des moyennes de 3 séries d'expériences indépendantes pour contrôler l'instabilité de l'optimisation (Clark *et al.*, 2011).

Sauf précision contraire les règles de réordonnancement sont filtrées selon un seuil de coût $max\_coût = 4$ et une longeur $max\_long = 10$.

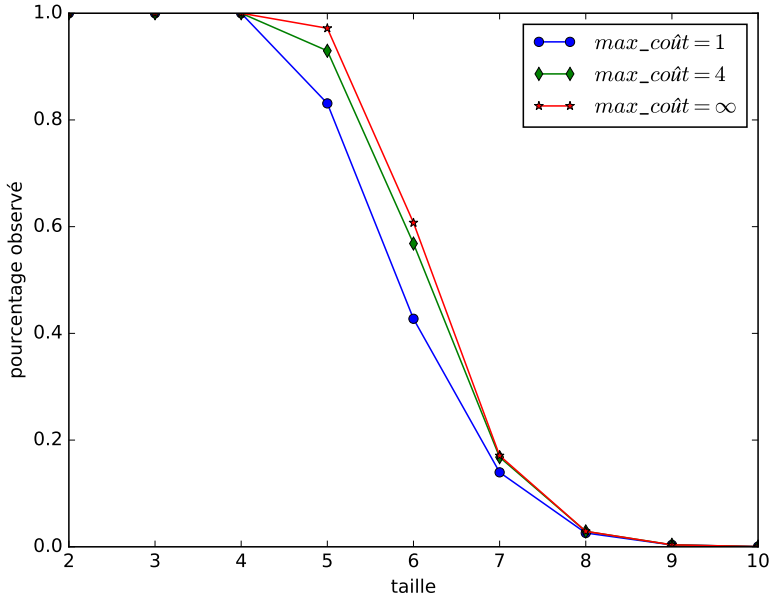
6.9 Couverture, généralisation et complexité de l'approche à base de règles

Une première question que l'on peut se poser concerne la couverture et la capacité de généralisation de notre système à base de règles. La figure 6.2 indique, pour chaque taille possible de réordonnancement n, le ratio entre le nombre de réordonnancements [14] observés dans les données par rapport au nombre total de réordonnancements possibles pour cette taille (i.e. r_n, défini à la section 6.4). Sans filtrage (avec $max\_coût = \infty$), tous les réordonnancements observés dans les données sont conservés. Dans ce cas, on s'aperçoit que quasiment toutes les permutations nécessaires sont observées jusqu'à une taille de 5. Ensuite, ce ratio décroît rapidement vers 0, dès que la taille des règles dépasse 10. De plus, on observe que la stratégie d'élagage ($max\_coût = 1$ ou 4) n'a ici qu'un faible impact négatif sur la couverture, et cela quelle que soit la direction de traduction envisagée.

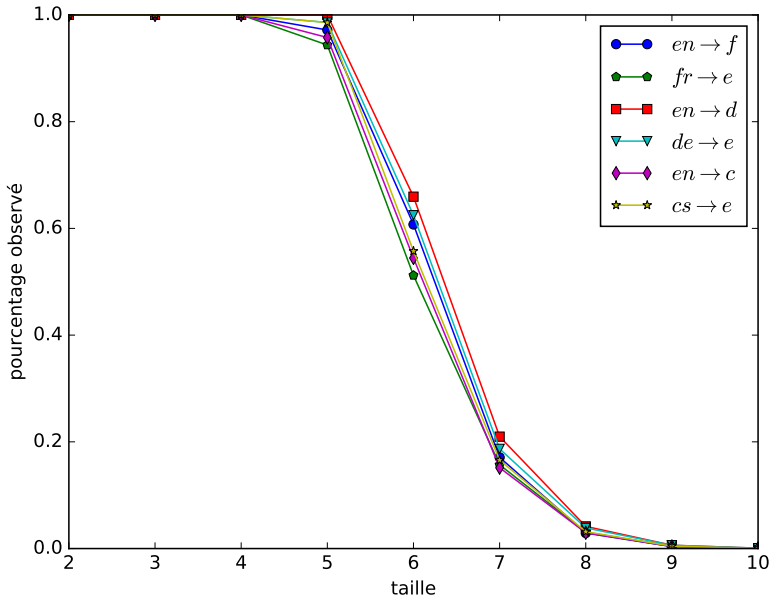
La figure 6.3 permet de caractériser la complexité des réordonnancements pour trois conditions différentes : ceux observés sur le corpus d'apprentissage, ceux nécessaires lors du test, et enfin ceux qui manquent lors du test [15], c'est-à-dire qui seraient nécessaires pour pouvoir réordonner une phrase source du test, mais pour lesquels aucune règle apprise ne permet cette permutation à cet endroit (*manque*). La figure 6.3 montre que la complexité des réordonnancements est approximativement égale dans les trois sous-populations considérées (apprentissage ; test ; manquant lors du test). On ne peut donc pas caractériser les réordonnancements qui sont mal capturés par l'approche à base de règles par une différence de complexité suivant nos critères. Dans le cas des réordonnancements de faible taille, on observe cependant un ratio d'ITG légèrement plus faible pour les réordonnancements manquants lors du test, un phénomène qui disparaît lorsque la taille augmente, cas où pratiquement tous les réordonnancements sont manquants, comme nous allons le voir maintenant.

14. On ne considère ici que les réordonnancements *minimaux* (§ 6.4), bien que l'on omette désormais l'adjectif.

15. Grâce ici à la nature minimale des réordonnancements, il est facile de voir qu'un réordonnancement (minimal) peut être capturé lors du test si et seulement si il existe précisément la règle pouvant s'appliquer aux positions de ce réordonnancement.

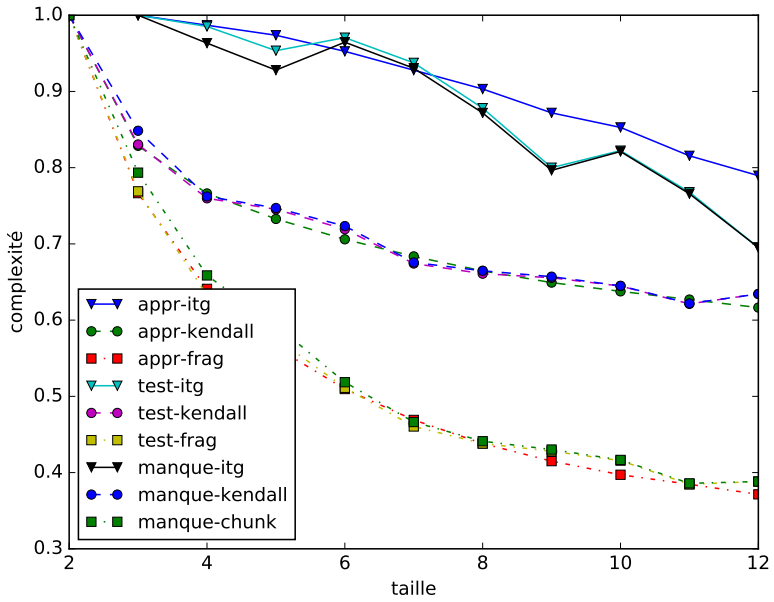


(a) $en \rightarrow fr$

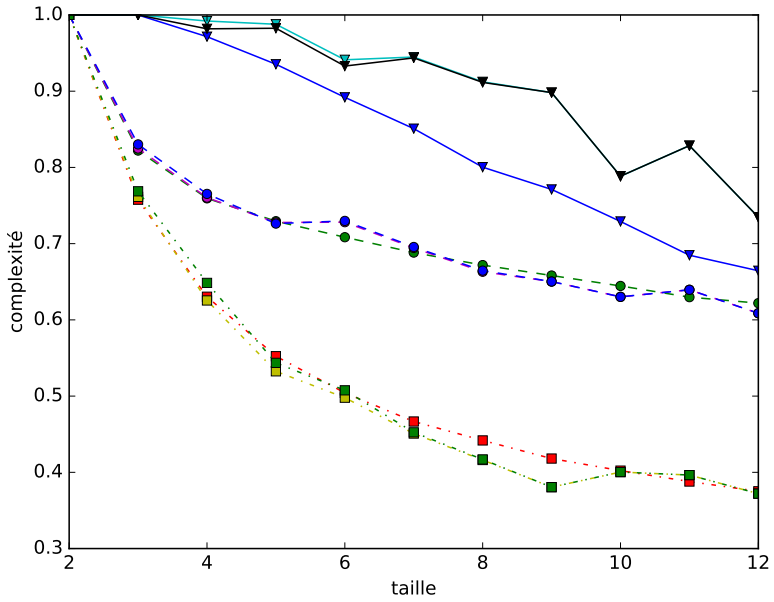


(b) $max\_coût = \infty$

FIGURE 6.2 – Pourcentage de réordonnancements observés sur les corpus d'apprentissage par rapport au nombre total de réordonnancements de cette taille, lorsque l'on utilise différents jeux d'étiquettes et différents seuils de filtrage ($max\_coût$) pour la direction $en \rightarrow fr$ (a) ou sans filtrage ($max\_coût = \infty$) pour toutes les directions (b). Les figures pour les cinq autres directions de traduction sont quasiment identiques à celle pour la direction $en \rightarrow fr$. Les réordonnancements de taille supérieure à 10 ne sont pas représentés, leur très faible ratio ne permettrait pas de se distinguer de 0.



(a) *en* → *de*



(b) *en* → *cs*

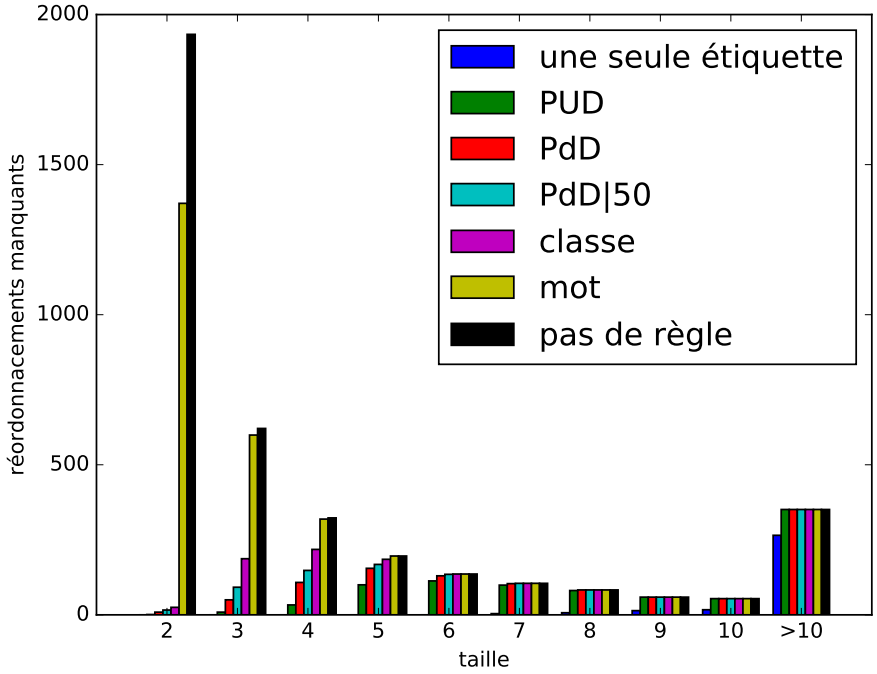
FIGURE 6.3 – Complexité des réordonnancements observés dans le corpus d'apprentissage (appr.), dans celui de test (test), ainsi que les réordonnancements observés en test mais qui ne seraient capturés par aucune règle (manque), en fonction de la taille des réordonnancements. La complexité est mesurée en indiquant le pourcentage de réordonnancements qui sont ITG (itg), ainsi que le τ de Kendall moyen (kendall) et de fragmentation (frag.) entre un réordonnancement et l'identité. Les graphes pour les quatre autres directions de traduction sont très similaires au cas *en* → *de*.

La figure 6.4 présente des histogrammes des réordonnancements manquants sur le corpus de test en fonction du jeu d'étiquettes utilisé pour calculer les règles de réordonnancement [16]. Pour la paire de langues anglais-français, la plupart des réordonnancements sont de longueur modérée et la moitié des réordonnancements ne s'étend que sur deux mots. Les réordonnancements impliquant de longues distances (i.e. plus de dix mots) sont peu fréquents, et la majorité d'entre eux provient d'erreurs lors de l'alignement forcé, d'erreurs de traductions, de traductions très éloignées d'une traduction littérale ou de constructions trop complexes pour un système de traduction automatique. En revanche, pour les paires de langues anglais-allemand et anglais-tchèque, la longueur des réordonnancements est répartie sur un plus grand nombre de valeurs, avec de nombreux réordonnancements de taille moyenne, mais également un nombre important de grands déplacements, qui ne peuvent pas être entièrement expliqués par des erreurs d'alignement.

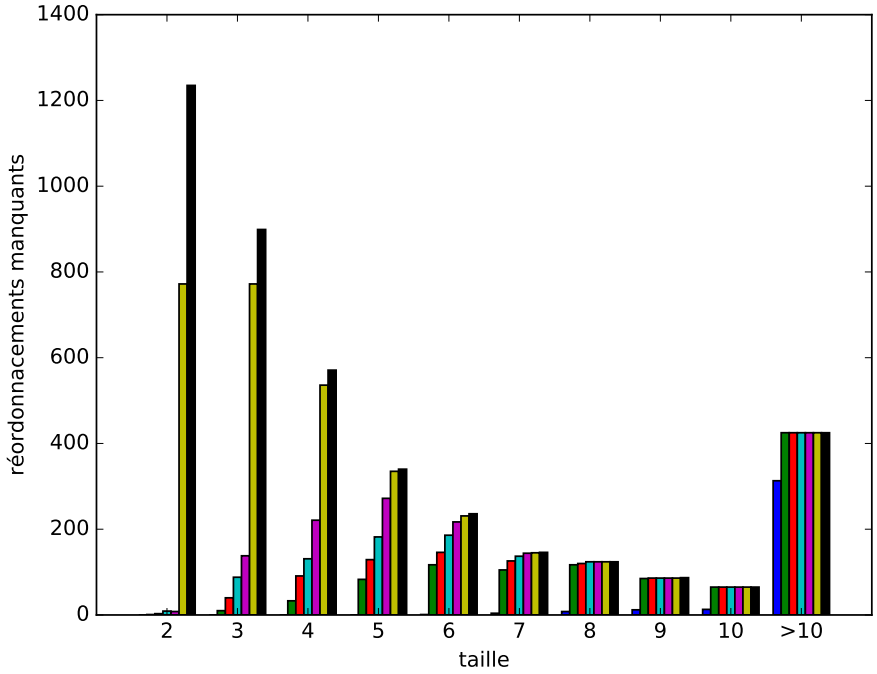
Même si seule une faible proportion des réordonnancements de grande portée est observée dans les corpus d'apprentissage, la figure 6.4 permet de confirmer que ces règles, au delà d'une taille de 10 mots, ne sont pratiquement jamais applicables sur les données de test et donc généralisent fort mal. En dessous de ce seuil, on remarque que les permutations observées sur les corpus d'apprentissage sont généralement suffisantes pour expliquer celles que l'on observe lors du test. Il est ainsi possible, pour les réordonnancements de taille moyenne (< 10 et > 5), de se restreindre aux permutations qui apparaissent dans le corpus d'apprentissage, en se rappelant que ces dernières ne représentent qu'une faible proportion de toutes les permutations possibles. Inversement, pour les règles de très faible portée (< 5), on s'aperçoit que toutes les permutations possibles sont observées dans le corpus d'apprentissage. Il semble alors intéressant d'utiliser des informations syntaxiques ou de contexte pour restreindre l'application des règles aux endroits les plus adaptés.

Enfin, la figure 6.4 nous permet d'y voir plus clair sur la puissance de généralisation des jeux d'étiquettes introduits dans la section 6.5.2. Avec des règles entièrement lexicalisées, la couverture sur le corpus de test est assez faible, plus de la moitié des inversions de mots serait manquée, et ce, pour toutes les directions de traduction considérées ici. Nous remarquons également que les règles à base de classes de mots sont systématiquement moins efficaces que lorsque l'on utilise un jeu d'étiquettes liées à des informations syntaxiques. Enfin, quel que soit le jeu d'étiquettes (en dehors du jeu réduit à une étiquette), plus de la moitié des réordonnancements de taille supérieure à 5 n'est pas atteignable par le système de règles. Ceci implique que dès que l'on cherche à restreindre l'application des règles, comme c'est le cas par la suite, l'approche ne fonctionne que pour les réordonnancements de taille relativement faible. Les différences entre les jeux d'étiquettes affectent

16. Le cas où aucune règle n'est appliquée indique donc le nombre de réordonnancements présents dans les données de test en fonction de leur taille.

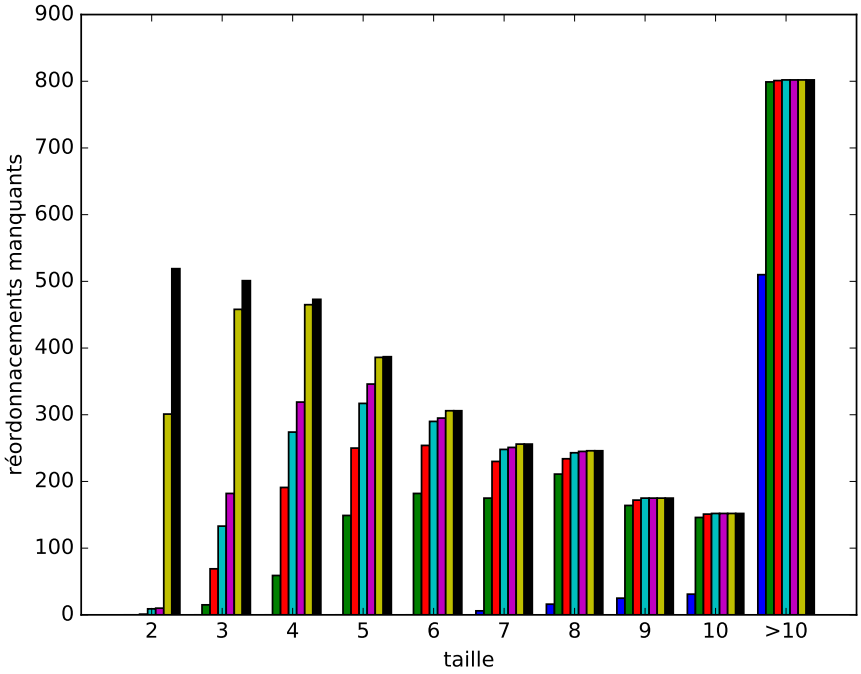


(a) $en \rightarrow fr$

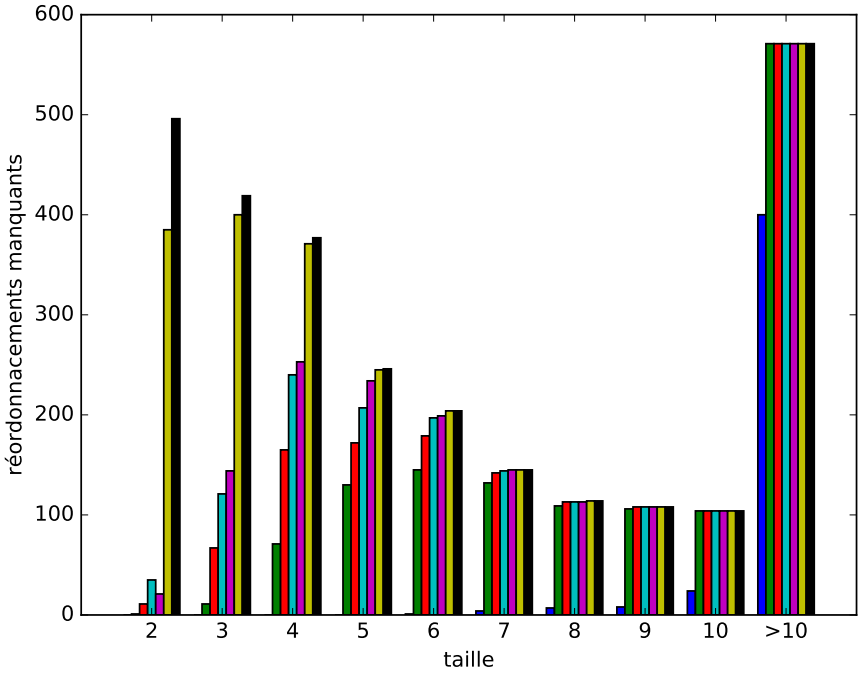


(b) $fr \rightarrow en$

FIGURE 6.4 – (suite p. 167) Nombre de réordonnancements manquants lors du test en fonction de leur taille lorsque que l'on utilise des règles construites à partir de différents jeux d'étiquettes (voir le texte pour plus de détails sur les jeux d'étiquettes utilisés, ici $max\_coût = \infty$) ou sans aucune règle (i.e. réordonnancements observés en test). Les histogrammes pour $de \rightarrow en$ et $cs \rightarrow en$ sont respectivement très semblables à ceux pour $en \rightarrow de$ et $en \rightarrow cs$.



(c) $en \rightarrow de$



(d) $en \rightarrow cs$

FIGURE 6.4 – suite de la page 166.

donc principalement ce qui se passe pour les mouvements à courte distance, pour lesquels ils offrent un compromis entre la couverture et la précision.

6.10 Conclusions

Dans ce chapitre nous avons étudié les différents réordonnancements qui pouvaient apparaître sur les corpus d'apprentissage, si l'on retrouvait ceux-ci sur un corpus de test et s'il était possible de les capturer par un système à base de règles sur des facteurs de mots. Dans le chapitre 7, nous poursuivrons cette étude en évaluant également l'impact des règles de réordonnancement et de différentes configurations sur les performances finales du système de traduction.

L'étude empirique des réordonnancements permet déjà de dresser plusieurs conclusions intéressantes. Dans l'étude que nous avons proposée, on peut répartir les réordonnancements en trois grandes familles, en fonction de leur taille :

- Les réordonnancements de courte distance (2-4 mots) : pour ces réordonnancements très locaux, on observe toutes les permutations possibles autant sur l'ensemble d'apprentissage que lors du test. Une approche à base de règles semble donc ici appropriée afin de permettre de contextualiser les réordonnancements et ainsi de construire un espace de recherche plus efficace. Nous avons vu que pour ces réordonnancements l'approche à base de règles fonctionnait bien, à condition d'utiliser des facteurs de mots suffisamment génériques.
- Les réordonnancements de portée moyenne (5-10 mots) : on observe que toutes les permutations ne sont pas observées et que les permutations observées lors du test l'ont également été lors de l'apprentissage. Ceci justifie donc l'approche à base de règles, qui permet de réduire de manière très importante la taille de l'espace de réordonnancement. Cependant, dès lors que l'on utilise des règles syntaxiques, à l'aide de facteurs de mots pour restreindre les positions d'applications des règles, on observe de sérieuses limites à l'approche par règle, qui ne permettent plus de généraliser, même lorsque l'on considère le jeu le plus général de facteurs de mots. Pour ces réordonnancements il semble donc intéressant de filtrer les permutations en conservant uniquement celles observées lors de l'apprentissage, sans nécessairement ajouter en plus de contraintes syntaxiques à base de facteurs sur les permutation autorisées.
- Les réordonnancements de grande taille (plus de 10 mots) : la grande majorité des permutations de grande taille est unique et trop rarement observée pour permettre de restreindre les permutations à celles observées précédemment. Les réordonnancements de grande distance ne sont donc pas capturés

par l'approche à base de règles, ce qui explique que l'on n'observe d'ailleurs pas de différence en limitant la taille de ceux-ci à 10. Pour les réordonnancements de grande portée, il est donc nécessaire de pouvoir généraliser au niveau des permutations, par exemple en définissant une classe de permutations (un sous-ensemble de permutations) englobant celles observées lors de l'apprentissage.

Au final, ces résultats suggèrent de décomposer davantage le problème du réordonnancement en fonction de la taille des permutations mises en jeu. Pour les réordonnancements de faible ou moyenne portée, on pourrait imaginer un système à base de règles dont la puissance de généralisation des étiquettes dépendrait de la taille du réordonnancement. Pour les réordonnancements de longue distance, notoirement difficiles à modéliser en traduction automatique, il conviendrait sans doute de se focaliser sur certains phénomènes particuliers.

Nous verrons dans le chapitre suivant que s'il est important de pouvoir mieux générer l'espace de recherche de permutation, la marge d'améliorations possibles n'est pas très importante, du moins dans le contexte des modèles actuels.

Chapitre 7

Importance de l'espace des réordonnancements

Sommaire

Dans ce dernier chapitre, nous évaluons l'importance de la qualité de l'espace des réordonnancements. Nous cherchons à comprendre si améliorer son design est susceptible d'influencer positivement les performances du système global de traduction, et dans quelle mesure, et si des études plus poussées doivent donc être menées pour améliorer ces espaces de recherche. Cette étude a été partiellement publiée dans (Pécheux *et al.*, 2014) et prolongée dans (Pécheux *et al.*, 2016a).

7.1 Introduction

En traduction automatique, les contraintes sur les mots, dont nous avons décrit certains modèles au chapitre précédent, ont une importance considérable sur l'ensemble des hypothèses de traductions qui peuvent être examinées pendant la recherche. L'espace de recherche de réordonnancement doit être conçu avec beaucoup de soin, en raison de contraintes computationnelles qui y sont directement associées, mais pas seulement. En effet, un espace plus grand contiendra potentiellement davantage de candidats, mais ces derniers peuvent également être noyés parmi bien d'autres hypothèses incorrectes et ainsi augmenter le risque d'erreurs de recherche, du fait de l'augmentation de l'ambiguïté et de l'interaction avec la recherche inexacte en faisceau. Dans ce chapitre, nous étudions l'importance du design de l'espace de réordonnancement, en utilisant le système état de l'art NCODE (§ 5.7), dans lequel tous les réordonnancements sont représentés dans un treillis de permutations avant le décodage. Cette approche en deux étapes nous permet d'étudier directement l'espace des réordonnancements et de mesurer son impact sur le processus complet de traduction. Cela nous permet également de comparer directement diverses manières de construire cet espace et d'étudier ses propriétés — sa taille, sa justesse, sa complétude —, et d'inclure également des analyses basées sur des oracles. Nous évaluons également le système à base de règles présenté au chapitre 6, en variant la longueur et le nombre de règles, le jeu d'étiquettes utilisé et contrastons cette approche avec des sous-ensembles purement combinatoires de permutations. Nous proposons des expériences pour les trois paires de langues, celles étudiées au chapitre 6, dans les deux directions, qui diffèrent du point de vue des réordonnancements à modéliser pour prendre en compte les différences structurales sous-jacentes. Nous considérons la paire anglais-français, une paire de langues relativement proches ; mais aussi anglais-allemand et anglais-tchèque, qui posent davantage de problèmes du point de vue de l'ordre des mots. Nous mesurons l'impact sur les performances effectives du système de traduction et réalisons des expériences oracles qui permettent de comprendre le potentiel éventuel des différents espaces dans le meilleur des cas, ainsi que l'influence des erreurs de recherche et des erreurs de modèles sur la qualité finale.

Les résultats expérimentaux révèlent qu'il reste encore du chemin à faire pour obtenir de meilleurs espaces de réordonnancement. Mais s'il est effectivement important de concevoir l'espace de recherche de réordonnancement de manière adéquate, nous montrons que les erreurs de recherche et les erreurs de modèles semblent être cependant un facteur d'erreurs encore plus important. Nous pensons que ces erreurs de recherche doivent être résolues en premier lieu si l'on veut pouvoir ensuite tirer profit de meilleurs espaces de réordonnancement. Des améliorations de l'espace de recherche doivent donc s'accompagner d'amélioration des modèles de réordonnancement, si l'on veut pouvoir observer des gains intéressants.

7.2 Modèles et contraintes, une histoire de compromis

Nous avons vu dans la section 6.1 que deux grandes questions sont soulevées lorsque nous nous intéressons au réordonnancement : le choix des réordonnancements possibles et la manière dont le modèle évalue les différentes alternatives. En réalité, il n'est pas possible d'établir une barrière nette entre ces deux problématiques, principalement du fait que même lorsque l'on utilise des contraintes strictes, la recherche exacte est rarement possible et qu'il est alors nécessaire d'utiliser des techniques de recherche approchée, qui conduisent à n'explorer qu'une partie de l'espace.

Dans ce chapitre, nous nous intéressons essentiellement au première problème, c'est-à-dire au choix de contraintes de réordonnancement appropriées. S'il est avéré que des améliorations des modèles de réordonnancement entraînent des améliorations conséquentes de la qualité des systèmes de traduction en permettant de choisir de meilleures hypothèses (Cherry, 2013; Auli *et al.*, 2014), il est moins évident de comprendre l'importance et l'impact des contraintes de réordonnancement sur l'ensemble du processus de traduction. En effet, au delà de considérations d'ordre computationnel, on est face à un compromis lors du choix de la construction de l'espace des réordonnancements qui peuvent être explorés pendant le décodage. D'une part, un espace suffisamment grand a plus de chance de contenir des permutations pouvant conduire à une meilleure traduction. D'autre part, à cause de l'ambiguïté des langues naturelles et d'une recherche inexacte sensible au nombre d'hypothèses, un espace conséquent peut également avoir pour conséquence davantage d'erreurs de recherche. Il est donc à la fois utile et important de mieux comprendre les limitations actuelles des espaces de réordonnancement en traduction. Parmi les questions que nous voulons poser dans ce chapitre, mentionnons :

- À quel point les espaces de recherche actuels sont-ils satisfaisants ?
- Comment les construire et quels sont les compromis à faire ?
- Est-il important que ceux-ci contiennent exactement les permutations nécessaires ou une bonne approximation de l'espace peut-elle être suffisante ?
- Les modèles de réordonnancement actuels sont-ils capables de profiter d'un meilleur espace de réordonnancement ?
- Dans quelle mesure une amélioration de l'espace de recherche de réordonnancement permettrait d'observer des gains substantiels en traduction ?

Pour répondre à ces questions et évaluer à quel point un système de traduction peut bénéficier d'un « meilleur » espace de réordonnancement, nous avons étudié différentes manières d'engendrer ces espaces et l'impact de ces derniers sur les performances globales. En étudiant également le cas d'une traduction monotone

et diverses conditions oracles, nous proposons également des bornes inférieures et supérieures sur le design possible des espaces de réordonnancement, éclairant également par là l'influence complexe des erreurs de recherche et de modèles sur la qualité finale.

Auli *et al.* (2009) s'intéressent également à l'importance de ce qu'ils appellent les *erreurs d'induction*, qui correspondent pour nous au cas où un réordonnancement nécessaire n'est pas dans l'espace de recherche, pour des modèles à base de segments et pour des modèles hiérarchiques. Ils proposent pour cela une mesure qui s'intéresse aux références atteignables ou non, similaire à ce que nous appellerons la *couverture* de l'espace de réordonnancement. Cependant, ces auteurs ne considèrent qu'une seule manière de faire varier l'espace de réordonnancement, c'est-à-dire variant la limite de distorsion, alors que nous étudions différentes manières d'engendrer les réordonnancements. Par contraste avec les travaux précédents, nous proposons de considérer simultanément plusieurs approches complémentaires pour mieux appréhender l'importance de l'espace de réordonnancement, en étudiant conjointement les performances effectives, les meilleures performances possibles (oracles) et les propriétés des espaces, ceci à la fois pour les approches à base de règles, pour des permutations purement combinatoires et pour des espaces de réordonnancement oracles.

Bisazza et Federico (2013b) soutiennent que le problème des réordonnancements à grande distance ne provient pas principalement de déficiences dans les modèles de réordonnancement, mais plutôt de la définition de l'espace de réordonnancement, trop large pour permettre au décodeur d'atteindre de tels réordonnancements. Ils introduisent à cet effet un modèle « mot après mot », semblable à celui de Visweswariah *et al.* (2011) pour *déformer* de manière dynamique l'espace de recherche pendant le décodage, afin de permettre ponctuellement l'utilisation de distorsions très élevées qui seraient impossibles sinon (Bisazza et Federico, 2013a). Avec leur approche, ils obtiennent un décodage accéléré, tout en observant des gains pour le réordonnancement des verbes dans une tâche de traduction de l'arabe vers l'anglais.

7.3 Métriques sur les espaces de réordonnancement

D'après nos hypothèses, le meilleur espace de réordonnancement, pour une phrase source donnée, devrait être l'espace le plus petit qui contient le réordonnancement (inconnu) de référence, donné par exemple en utilisant la procédure de démêlage (cf. § 5.7). Ceci nous conduit donc à proposer comme mesure de qualité, pour évaluer une certaine classe de contraintes de réordonnancement, la

couverture sur un ensemble de test, c'est-à-dire le nombre de fois où l'espace de réordonnancement contient le réordonnancement de référence. Parallèlement, nous calculons également la *taille* de l'espace, donnée par le nombre de chemins [1] et donc de permutations encodées. Nous utilisons également le nombre d'arcs du treillis, étant donné que celui-ci est directement lié à la complexité du décodage.

7.4 Des réordonnancements oracles

Lors de l'apprentissage, les phrases sources sont réordonnées de manière déterministe, en utilisant la procédure de démêlage, rendant ainsi possible l'extraction des tuples et l'estimation des modèles. Lors du décodage, l'idéal serait de pouvoir avoir directement accès à ce réordonnancement de référence, et ainsi de pouvoir présenter les mots de la phrase source dans un ordre similaire à celui de la phrase cible, donné par la permutation issue de la procédure de démêlage. Pour les phrases de test, il est possible d'utiliser pour cela une stratégie « oracle » au sens où l'on prend en compte la traduction de référence pour construire cet alignement, à l'aide des liens d'un alignement forcé entre la phrase source et sa référence. Ainsi, le meilleur espace de réordonnancement serait celui qui ne contiendrait comme unique alternative que ce réordonnancement de référence, ce qui est naturellement impossible en pratique puisque cela imposerait déjà de connaître la référence attendue. Nous appelons ce réordonnancement oracle *réordonnancement de référence* [2].

7.5 Quel est le meilleur réordonnancement atteignable ?

En un certain sens, le réordonnancement de référence peut être considéré comme le meilleur réordonnancement possible que l'on puisse espérer. Cependant, pour de nouvelles phrases, ce réordonnancement oracle peut nécessiter des réordonnancements de longue distance ou des permutations rares qui ne sont pas observées dans les données d'apprentissage. Dans nos expériences (§ 7.6.5), seulement environ 20 à 60% des réordonnancements de référence sont effectivement atteignables par notre système à base de règles, selon la direction de traduction considérée et la configuration utilisée. Il est donc intéressant d'étudier également les propriétés du « meilleur » réordonnancement que le système peut engendrer, en plus de celui d'un réordonnancement qui peut être relativement artificiel.

1. Calculé de manière efficace en utilisant le semi-anneau de comptage.

2. Nous préférons éviter par la suite la dénomination de réordonnancement oracle pour ne pas introduire de confusions avec le décodage oracle (§ 7.5.1), qui utilise également la référence pour choisir la meilleure hypothèse dans l'espace de recherche.

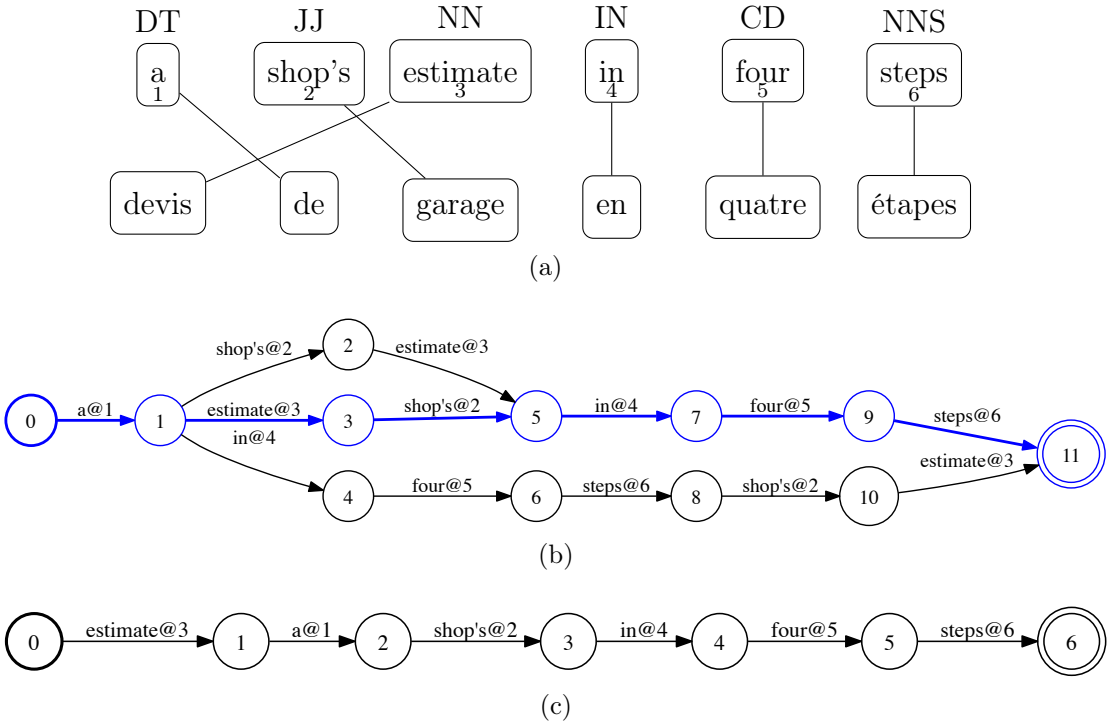


FIGURE 7.1 – Phrase source annotée de la figure 6.1, avec la phrase cible correspondante et l'alignement forcé entre les deux (a) ; le treillis de réordonnancement que l'on obtient lorsque l'on applique ces règles, où le meilleur réordonnancement défini à la section 7.5 est en bleu gras (b) ; le treillis de réordonnancement de référence construit à partir du démêlage des liens d'alignement (c).

Le meilleur réordonnancement peut être par exemple défini comme étant la permutation du treillis de réordonnancement qui conduit à la meilleure hypothèse de traduction. Une telle définition impose cependant une dépendance envers le système de traduction complet, y compris la stratégie de recherche approchée et les modèles de traduction. Au lieu de cela, nous reprenons l'idée de Herrmann *et al.* (2013b) en définissant le « meilleur » réordonnancement comme étant le plus proche, en un sens à définir, du réordonnancement de référence. Cette approximation suppose que le meilleur ordre est celui qui ressemble le plus à l'ordre de la traduction de référence, ce qui est relativement raisonnable étant donné que les métriques automatiques que nous utilisons pour évaluer nos systèmes reposent sur une hypothèse semblable.

Définir la permutation la plus proche demande de choisir une métrique sur les permutations. Parmi les nombreux choix possibles (Deza et Huang, 1998), deux métriques se sont révélées être particulièrement utiles pour évaluer la qualité d'un

réordonnancement : le τ de Kendall (Isozaki *et al.*, 2010a; Birch *et al.*, 2010; Talbot *et al.*, 2011; Neubig *et al.*, 2012) et la distance de fragmentation (en anglais *fragmentation chunk*, dite aussi *fuzzy reordering*) (Banerjee et Lavie, 2005; Talbot *et al.*, 2011; Neubig *et al.*, 2012). Nous avons introduit ces deux métriques à la section 6.9. Dans ce travail, nous utilisons le τ de Kendall qui corrèle bien avec les jugements humains (Birch *et al.*, 2010). Rappelons que cette métrique revient à compter le nombre d'inversions entre deux permutations.

Nous expliquons par la suite comment il est possible de trouver efficacement la permutation la plus proche d'une permutation de référence en terme de distance du τ de Kendall dans un treillis. Observons d'abord que

$$\tau(\sigma, \pi) = \tau(\pi^{-1} \circ \sigma, id) \tag{7.1}$$

où id est la permutation identité. Quitte à réindexer, le problème est alors de trouver la permutation dans un treillis $\mathcal{L}$ comprenant le nombre minimal d'inversions

$$\arg\min_{\sigma \in \mathcal{L}} \; inv(\sigma) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathbb{1}_{\{\sigma_i > \sigma_j\}} = \sum_{j=1}^{n} \sum_{i=1}^{j-1} \mathbb{1}_{\{\sigma_i > \sigma_j\}} = \sum_{j=1}^{n} w(\sigma_j, \{\sigma_i\}_{i=1}^{j}) \tag{7.2}$$

où $w(k, S) = \sum_{s \in S} \mathbb{1}_{\{s>k\}}$ compte combien de fois un entier k est inférieur à un élément d'un ensemble d'entiers $S \in 2^n$. Le nombre d'inversions se décompose comme une somme de fonctions locales qui ne dépendent que de *l'ensemble* des indices déjà permutés. Comme nous l'avons observé au chapitre précédent (§ 6.6), dans un treillis de permutations $\mathcal{L} = (V, E, \Sigma, w)$, à chaque noeud $v \in V$ correspond un ensemble d'entiers S_v. Chaque arc $e \in E$ quittant v avec comme étiquette un entier k va avoir une contribution $w(k, S_v)$ au nombre total d'inversions de n'importe quel chemin du treillis passant par e. Ainsi, on peut pondérer $\mathcal{L}$ avec les poids $w(k, S_v)$ et utiliser l'algorithme classique de plus court chemin pour calculer le meilleur réordonnancement.

La figure 7.1 montre un exemple de treillis de réordonnancement, avec trois chemins différents — distants (au sens du τ de Kendall) respectivement de 1, 2 et 8 du réordonnancement de référence — et met en avant le « meilleur » chemin en gras et en bleu.

Comme de nombreux chemins dans le treillis peuvent atteindre la plus courte distance avec le réordonnancement de référence (c'est-à-dire que le « arg min » de l'équation (7.2) n'est pas forcément unique), le plus court chemin est en fait un sous-treillis du treillis original, comprenant tous ces chemins. Dans nos expériences, nous observons cependant que les treillis de meilleurs réordonnancements se composent généralement d'un seul chemin, puisqu'ils comportent, en moyenne, 1.1 chemins.

7.5.1 Analyse oracle

Nous utiliserons également une analyse oracle des performances. Les expériences oracles offrent une méthode fort intéressante pour analyser divers aspects des systèmes de traduction automatique. Par exemple, elles permettent d'identifier les erreurs de traduction dans les tables de bisegments (Wisniewski *et al.*, 2010), ou d'effectuer des analyses d'erreurs (Wisniewski et Yvon, 2013). Sokolov *et al.* (2012) décrivent différentes méthodes efficaces pour trouver la meilleure hypothèse en BLEU dans un treillis de recherche et utilisent leurs algorithmes pour comparer les espaces de recherche explorés par MOSES et par NCODE. Dans notre travail, nous calculons également les oracles, mais sur les treillis (de réordonnancement) *complets*, c'est-à-dire avant élagage par la méthode de recherche en faisceau. Une autre application, proche de notre objectif, est d'étudier les limites imposées par différentes contraintes de réordonnancement. Dreyer *et al.* (2007) calculent une borne inférieure du meilleur BLEU qu'il est possible d'atteindre en utilisant des techniques de programmation dynamique pour les contraintes IBM et ITG. Sokolov *et al.* (2012) montrent un effet très limité sur l'influence de la limite de distorsion tant pour les performances réelles que leurs analyses oracles. Wisniewski et Yvon (2013) explorent de manière plus systématique différentes contraintes de réordonnancement, en particulier la limite de distorsion et les contraintes IBM et MJ. Nous verrons que nous partageons avec ce travail leurs conclusions principales : la fonction de score (i.e. les modèles) semble être la principale limitation pour les systèmes à base de segments, alors même qu'ils sont suffisamment expressifs pour atteindre de bien meilleurs résultats, si l'on considère la qualité de l'espace de recherche. Quant à considérer des réordonnancements oracles, Khalilov et Sima'an (2012) introduisent une borne supérieure, similaire à nos réordonnancements de référence, et montrent qu'il reste une marge importante à combler pour les méthodes de préordonnancement, bien que celle-ci soit limitée si l'on se restreint uniquement à des contraintes issues de permutations des noeuds d'arbres d'analyse syntaxique.

Nous calculons les oracles comme expliqué dans la section 5.5, en utilisant ici systématiquement les paramètres $p = 0.4$, $r = 0.8$ et $\theta_0 = -1$ qui ont donné un bon compromis de résultats dans nos expériences préliminaires.

7.6 Expériences

Nous utilisons exactement le même cadre expérimental qu'au chapitre 6, tel que décrit à la section 6.8.1.

```
   src : the meeting was announced by the president's spokesman Radim Ochvat .
   ref : c' est le porte-parole présidentiel Radim Ochvat qui a informé de la réunion .
  mono : la réunion a été annoncée par le président porte-parole Radim Ochvat .
règles : la réunion a été annoncée par le porte-parole du président Radim Ochvat .
déplié : le porte-parole du président Radim a été annoncée par la réunion Ochvat .
oracle : de la réunion est le porte-parole présidentiel Radim Ochvat qui
```

FIGURE 7.2 – Traduction d'une phrase source (src) de NEWSTEST2010, sa traduction de référence (ref), et les hypothèses données par des décodages monotones (mono) ; utilisant les treillis de réordonnancement de règles (règles) ; ou le réordonnancement de référence (déplié). La dernière ligne est l'hypothèse choisie par le décodage oracle (oracle) dans l'espace de recherche de traduction. Pour cet exemple, le décodage en utilisant le treillis de meilleur réordonnancement ou sa combinaison avec le réordonnancement oracle proposent la même hypothèse que pour le treillis de règles.

7.6.1 D'un décodage monotone aux treillis de réordonnancement : impact sur les performances

Dans cette section, nous étudions l'influence de l'espace de réordonnancement et cherchons à comprendre où se situe l'espace restreint obtenu par l'application des règles de réordonnancement, en particulier par rapport à des bornes inférieures (monotone) ou supérieures (réordonnancements de référence), en terme de performances du système de traduction global. Dans cette optique, le tableau 7.1 montre les résultats obtenus en terme de métrique BLEU sur le corpus de test pour des espaces de réordonnancement de « qualités » différentes. Le premier espace de réordonnancement possible, le plus simple, ne comprend que l'ordre initial de la phrase source, autrement dit, la permutation identité, ce qui revient à considérer un décodage monotone (monotone). Le deuxième espace utilise notre approche à base de règles pour engendrer les treillis de réordonnancement (règles). Les quatre autres espaces proviennent de configurations oracles qui seront détaillées dans les sections 7.6.2 et 7.6.3. Un exemple de traduction d'une phrase source selon ces différentes configurations est représenté par la figure 7.2. Le tableau 7.1 expose également les meilleurs scores BLEU possibles (obtenus par un décodage forcé) pour les six conditions considérées.

Pour les paires de langues anglais-français et anglais-allemand, nous observons, comme on pouvait s'y attendre, un gain en BLEU lorsque l'on passe d'un décodage monotone à un décodage où les réordonnancements de notre système à base de règles peuvent être utilisés. Pour anglais-français, le gain peut atteindre jusqu'à 3 points BLEU, ce qui montre l'importance de prendre en compte les mouvements des mots pendant le processus de traduction, même pour des langues relativement proches. Sur la figure 7.2 on voit par exemple que la traduction monotone ne

	cal.	déc.	$en \rightarrow fr$	$fr \rightarrow en$	$en \rightarrow de$	$de \rightarrow en$	$en \rightarrow cs$	$cs \rightarrow en$
NCODE	*règles*	*mono*	19.1± 0.0	19.8± 0.0	12.3± 0.0	17.0± 0.0	9.9± 0.0	14.7± 0.0
	règles	*règles*	22.2± 0.0	21.9± 0.0	12.7± 0.0	18.1± 0.0	9.9± 0.0	14.8± 0.0
	règles	*meilleur*	22.8± 0.0	24.2± 0.0	13.5± 0.0	19.0± 0.0	10.2± 0.0	15.3± 0.0
	règles	*déplié*	24.0± 0.0	25.8± 0.0	15.6± 0.0	22.1± 0.0	10.9± 0.1	16.4± 0.1
	règles	*aug*	22.3± 0.0	21.9± 0.0	12.8± 0.0	18.6± 0.1	9.9± 0.0	14.8± 0.0
	règles	*duel*	23.1± 0.0	24.5± 0.0	14.1± 0.0	20.2± 0.0	10.3± 0.0	15.5± 0.0
Oracle		*mono*	47.0	50.0	37.8	43.0	30.1	36.2
		règles	54.2	57.5	42.8	49.0	33.1	40.2
		meilleur	52.5	56.2	40.7	46.8	31.7	38.0
		déplié	54.8	59.2	45.0	52.4	34.7	40.4
		aug	56.0	59.9	46.1	53.4	35.5	42.0
		duel	54.9	59.3	45.2	52.6	35.0	40.8
NCODE	*aug*	*aug*	22.3± 0.0	22.0± 0.0	13.7± 0.0	19.9± 0.1	10.1± 0.0	14.9± 0.0
	duel	*duel*	23.9± 0.0	25.6± 0.0	15.6± 0.0	21.9± 0.0	11.2± 0.0	16.6± 0.0
	aug	*règles*	22.2± 0.0	22.0± 0.0	12.1± 0.1	17.6± 0.2	10.0± 0.0	14.8± 0.0

Tableau 7.1 – Scores BLEU pour le système NCODE et pour un décodage oracle, lorsqu'aucun réordonnancement n'est autorisé (monotone (mono)) ; en utilisant l'espace engendré par les règles de réordonnancement (règles) ; lorsque l'on ne considère que le treillis de meilleur réordonnancement (meilleur) ; lorsque l'on donne exactement le réordonnancement de référence (déplié) ; lorsque l'on ajoute ce réordonnancement de référence au treillis de règles (aug) ; et enfin lorsque l'on regroupe en compétition le treillis de meilleur réordonnancement et le réordonnancement de référence (duel). Les résultats sont donnés pour le corpus de développement (cal.) et lorsque l'on décode le test (déc.). Les scores présentés sont les moyennes de 3 calibrages indépendants avec MIRA et les déviations standard entre les résultats des expériences sont rapportées en petites lettres.

permet pas d'inverser `president's` et `spokesman`, ce qui aboutit à une traduction incorrecte. Pour anglais-allemand, le gain est bien plus faible, en particulier pour la direction $en \rightarrow de$, où l'amélioration est de seulement un demi point BLEU. Ceci semble suggérer que notre système de réordonnancement n'est pas capable de bien engendrer ou de prédire le bon ordre des mots lorsque l'on traduit vers l'allemand. Enfin, et de manière plus surprenante, nous n'observons aucun gain pour la paire de langues anglais-tchèque, et ce quelle que soit la direction, ce qui indique que notre

système de règles n'est peut être pas adapté pour capturer les traductions depuis et vers une langue où l'ordre des mots est assez libre. Deux explications peuvent être avancées pour comprendre ce résultat négatif. Soit notre mécanisme de réordonnancement n'est pas assez expressif pour être capable d'engendrer des variantes de réordonnancement intéressantes, soit les modèles de réordonnancement (et de traduction) ne sont pas en mesure de reconnaître les meilleurs chemins dans les treillis de réordonnancement. Nous verrons en effet à la section 7.6.5 (tableau 7.2) que le bon ordre n'est présent que pour 30 à 40% des phrases. Cependant, les décodages oracles montrent que dans tous les cas, et ce même pour les directions de traduction qui représentent le plus de difficultés, les treillis de réordonnancement obtenus à partir des règles contiennent de meilleurs chemins qui pourraient être exploités par le décodeur pour obtenir un meilleur score BLEU. Ceci montre que les erreurs de modèle ou de recherche sont en bonne partie responsables des difficultés à obtenir des gains lorsque l'on considère des espaces de réordonnancement pourtant plus riches.

Le tableau 7.1 nous apprend en outre que les résultats en BLEU pour $en \rightarrow de$, et encore plus pour $en \rightarrow cs$, sont inférieurs aux autres, ce qui suggère que ces directions de traduction sont plus complexes à traiter [3]. En effet, l'allemand et le tchèque sont deux langues morphologiquement riches qui font souvent appel à des réordonnancements de grande portée (ce qui était mis en évidence par la figure 6.4) lorsque l'on traduit à partir de l'anglais.

Les scores élevés que l'on obtient lors du décodage forcé suggèrent également que des gains plus importants sont à attendre de l'amélioration des modèles de traduction (et de réordonnancement) plutôt qu'en améliorant les espaces de recherche de réordonnancement. Il ne faut cependant pas oublier que les scores en BLEU du décodage oracle peuvent être largement optimistes et qu'une part des gains observés peut être attribuée à un surapprentissage des particularités de la métrique BLEU. Une illustration de ce problème est donnée par la figure 7.2, sur laquelle on peut observer la piètre qualité de l'hypothèse de décodage oracle.

7.6.2 Réordonnancements oracles : une borne supérieure sur les performances

Afin de mieux comprendre l'impact des erreurs de modèle et de recherche, nous avons effectué plusieurs expériences complémentaires avec deux treillis de réordonnancement de contrôle. Les résultats sont collectés dans le tableau 7.1. La configuration *meilleur* revient à utiliser le treillis de meilleur réordonnancement que nous avons défini dans la section 7.5 tandis que le cas *déplié* correspond à l'utilisation

3. Rappelons qu'ici les ensembles de test sont les mêmes pour toutes les langues, ce qui permet la comparaison.

directe du réordonnancement de référence. Le tableau 7.1 illustre le fait qu'il y a un net avantage, pour toutes les directions de traduction, à connaître directement le meilleur réordonnancement, permutation qui était pourtant présente dans les treillis de réordonnancement. Le gain est en particulier assez important pour la direction $en \rightarrow fr$. Ceci montre bien que les erreurs de recherche et de modèle sont responsables du choix d'un chemin de réordonnancement moins avantageux dans le cas où un treillis plus fourni est utilisé, et permet de quantifier l'impact des erreurs de modèle et de recherche pendant le décodage.

Il faut cependant remarquer, et cela a également été observé par Herrmann *et al.* (2013b), que le « meilleur » réordonnancement dans le treillis tel que nous l'avons défini n'est pas nécessairement celui qui conduit à la meilleure traduction possible. En effet, des erreurs d'alignements peuvent se traduire par un réordonnancement de référence (après démêlage) de qualité douteuse, ce qui affectera à son tour l'approximation qui en est faite : le meilleur réordonnancement. De plus, l'ordre ainsi obtenu peut être dans certains cas un peu artificiel et ne pas permettre parfaitement de s'ajuster correctement au mécanisme de segmentation en tuples[4]. Le décodage oracle permet de donner une évaluation quantitative de ce problème, car le score BLEU du décodage oracle lorsque l'on utilise directement le treillis de meilleur réordonnancement se trouve à mi-chemin entre le cas monotone et celui du treillis complet de règles. Ceci signifie que dans de nombreux cas, une meilleure permutation que celle(s) du treillis de meilleur réordonnancement, pourrait être choisie et conduire à de meilleures performances. Inversement, étant donné que l'on utilise la référence dans le calcul du réordonnancement de référence, et que l'on a donc une connaissance relative de l'ordre de ses mots, on peut penser que le décodage est avantageusement biaisé vers cette référence. Ceci peut conduire l'évaluation, qui est faite en prenant en compte cette même référence, à être plus optimiste que ce que seraient les performances réelles d'un système. Cette expérience comporte donc deux biais antagonistes, mais nous pensons que si les résultats quantitatifs doivent être considérés avec prudence, l'interprétation qualitative qui s'en dégage illustre bien notre propos.

L'expérience précédente fournit donc une sorte de borne supérieure quant aux meilleures performances possibles d'un système étant donné un espace de réordonnancement, ici celui contraint par des règles de réordonnancement. Il serait également intéressant de contraster ce résultat – celui du meilleur score atteignable pour cet espace de réordonnancement si on savait choisir le bon chemin – avec la meilleure performance possible que l'on obtiendrait avec le meilleur espace de réordonnancement possible. Ceci nous donnerait ainsi de précieuses indications sur la qualité de l'approximation d'un espace de recherche parfait par notre système de

4. Même si ce problème ne semble pas très important dans notre cas, du fait du choix de segments (tuples) minimaux dans Ncode.

règles (Herrmann *et al.*, 2013b). De plus, cela fournirait également une borne supérieure sur n'importe quel mécanisme envisageable qui serait utilisé pour engendrer l'espace de réordonnancement.

Comme le montre clairement le tableau 7.1, toutes les directions de traduction sont nettement avantagées par la connaissance du réordonnancement de référence, avec des résultats qui peuvent être importants, malgré une certaine disparité selon les langues. La différence entre la condition avec le meilleur réordonnancement et celle avec le réordonnancement de référence mesure l'écart dû à un ensemble de contraintes de réordonnancement imparfaites, en se plaçant dans le cas où il n'y a aucune erreur de modèle ou de recherche pour le choix du réordonnancement. Inversement, la différence entre la configuration comprenant le treillis de règles en totalité et celle où le réordonnancement de référence est donné mesure les gains possibles si l'on était capable à la fois de construire un meilleur espace de réordonnancement *et* de trouver le meilleur chemin dans cet espace.

Un gain de presque 4 points BLEU lorsque l'on traduit de l'anglais vers le français ou l'allemand montre qu'il reste encore du travail à faire. Cependant, l'amélioration pour la direction $en \rightarrow cs$ n'est pas immédiate. En d'autre termes, « résoudre » le problème du réordonnancement au moment du décodage n'a qu'un effet moindre sur les performances pour cette direction de traduction. Pour cette paire de langues, il semble donc que le choix des contraintes que l'on pose sur l'espace de réordonnancement ne soit pas le principal facteur limitant du système de traduction. Il ne faut d'ailleurs pas oublier que, dans le cas du réordonnancement de référence, il n'y a pas de limite de longueur sur la taille des réordonnancements minimaux qui le composent, ce qui écarte le manque de réordonnancement de grande portée comme explication possible des différences observées.

7.6.3 Discrimination du réordonnancement de référence

Pour consolider les observations précédentes, nous avons conçu deux expériences additionnelles qui consistent à mettre en compétition certains des treillis de réordonnancement précédents.

En premier lieu, nous enrichissons le treillis de réordonnancement donné par l'approche à base de règles en ajoutant comme chemin additionnel le réordonnancement de référence (ligne étiquetée par (*aug*) dans le tableau 7.1). Pour cette configuration, l'espace de réordonnancement contient ainsi le réordonnancement souhaité ainsi qu'un grand nombre de compétiteurs fallacieux. Cette expérience nous permet de mieux comprendre dans quelle mesure le décodeur est effectivement capable de choisir le réordonnancement de référence dans le treillis. Le tableau 7.1 montre qu'il n'y a quasiment aucune différence avec le treillis de règles original, exception faite de la direction $de \rightarrow en$. Ceci confirme donc que l'un des principaux problèmes avec le réordonnancement n'est pas la question de savoir si l'espace de

recherche contient ou non le bon réordonnancement, mais bien de savoir si le système est capable de le reconnaître et de l'utiliser. On souligne donc à nouveau l'importance des erreurs de recherche et de modèle.

Il est vrai cependant que le réordonnancement de référence n'est pas forcément toujours celui qui engendrerait la meilleure traduction. Par exemple sur la figure 7.2, la traduction issue du treillis de règles est légitime, bien que différente de la traduction de référence. En revanche, la traduction donnée en partant du réordonnancement de référence aboutit à un contresens, en raison de la construction à la voix passive, complexe pour le système de traduction automatique. Le décodage oracle permet de confirmer cette observation, tout en lui conférant une interprétation plus quantitative : les résultats du tableau 7.1 pour le décodage oracle avec le treillis augmenté sont toujours supérieurs à ceux obtenus en ne considérant que le réordonnancement de référence. Ceci implique que, dans de nombreux cas, l'espace de recherche contient un réordonnancement qui permet d'atteindre une meilleure traduction que celle donnée en utilisant le réordonnancement de référence.

La deuxième expérience propose une comparaison entre le réordonnancement de référence et le(s) meilleur(s) réordonnancement(s) du treillis de règles. Pour chaque phrase source, on considère ainsi un treillis de permutations regroupant le treillis de meilleur réordonnancement et le réordonnancement de référence (ligne notée (*duel*) dans le tableau 7.1). Cette expérience vise à mesurer à quel point le décodeur serait capable de choisir le réordonnancement de référence [5] en l'absence de l'ambiguïté instaurée par les nombreux autres compétiteurs du treillis de règles, comme c'était le cas dans l'expérience précédente. Le tableau 7.1 indique que pour les paires de langues anglais-français et anglais-tchèque le score obtenu est à peine meilleur que celui de la configuration utilisant le seul meilleur réordonnancement, ce qui implique que le décodeur n'est pas amené à choisir le réordonnancement de référence à moins d'y être forcé et illustre un cas concret d'erreur de modèle (il n'y a pas d'erreur de recherche dans ce cas [6]). Une explication possible serait que dans certains cas le réordonnancement de référence pourrait contenir de nombreux réordonnancements de grande taille, qui seraient sévèrement « punis » par le modèle de distorsion et ne pourraient donc pas être choisis lors du décodage. Dans tous les cas, ceci met en lumière une difficulté du côté des modèles utilisés pour évaluer les hypothèses. La différence est minimale pour la paire de langue anglais-tchèque, paire de langues pour laquelle il semble que la fonction de score soit la moins appropriée, au vu des expériences précédentes.

5. Du moins dans les cas où celui-ci permet de trouver une meilleure hypothèse. En fait, le décodage oracle (tableau 7.1) nous permet de voir que dans certains cas le meilleur réordonnancement peut être plus approprié que le réordonnancement de référence, car le score oracle de la configuration jointe est meilleur que celui des configurations prises isolément.

6. L'espace de recherche étant réduit à quelques hypothèses seulement, la recherche par faisceau est ici exacte.

Hormis dans l'expérience précédente, nous n'avons pas essayé d'isoler les erreurs de recherche des erreurs de modèle. Une perspective intéressante serait d'évaluer la couverture et les oracles sur les espaces de recherche qui sont *réellement* explorés par le décodeur, c'est-à-dire *après* élagage par la recherche en faisceau, ce qui permettrait de mieux séparer la contribution des deux types d'erreurs. La dernière expérience de compétition entre deux réordonnancements montre cependant que les erreurs de modèle jouent un rôle important, étant donné qu'il n'y a pas d'erreur de recherche dans l'espace extrêmement réduit de cette configuration.

7.6.4 Choix de l'espace de réordonnancement lors du calibrage

Enfin, nous avons également étudié l'importance du choix de l'espace de réordonnancement pendant la phase de calibrage des poids du modèle. Le tableau 7.1 montre qu'utiliser le treillis augmenté du réordonnancement de référence lors du test permet d'améliorer les performances si l'on utilise également cet espace lors du calibrage. Cela suggère que l'étape de calibrage sait tirer parti du fait d'avoir le réordonnancement de référence comme un candidat possible. Cet effet est encore plus important lorsque l'on effectue le calibrage et le décodage sur le treillis de duel de la section précédente, avec un gain d'environ un point BLEU quelle que soit la direction. Ceci nous amène à mesurer nos conclusion précédentes : le décodeur est dans certains cas capable de reconnaître le réordonnancement de référence, à condition que ce soit également le cas lors du calibrage des modèles [7].

Malheureusement, bien qu'il soit aisé de calibrer les modèles en utilisant le treillis augmenté du réordonnancement de référence, ceci n'est plus possible en pratique lors du test où l'on ne peut utiliser — hors expériences oracles — que le treillis de règles. Ce dernier scénario n'est pas intéressant puisque, comme le montre la dernière ligne du tableau 7.1, il conduit à de moins bons résultats que le calibrage sur le treillis de règles.

7.6.5 Compromis sur l'espace de réordonnancement

Le tableau 7.2 donne la taille, la couverture et les scores obtenus lors des décodages standard et oracles de différents espaces de réordonnancement, en faisant varier le paramètre de seuil dans le filtrage des règles de réordonnancement. On

7. Comme nous l'avons déjà suggéré ci-dessus, la pénalité de distorsion est largement responsable de cela. En effet, lorsque l'on calibre sur le corpus de développement avec le treillis de règles, le poids de distorsion est de 0.01, pénalisant donc légèrement les réordonnancements. En revanche, il est respectivement de -0.05 et -0.19 lorsque l'on utilise le treillis augmenté ou le treillis de duel lors du calibrage, *encourageant* ainsi des écarts à l'ordre monotone.

	max_coût	BLEU NCODE	BLEU oracle	#règles	taille	couverture (%)
$en \rightarrow fr$	0	19.1	47.0	0k	27 / 1	20
	2	22.0	52.4	23k	34 / 45	41
	4	22.2	54.2	33k	52 / 10^5	51
	∞	21.9	57.5	42k	10^2 / 10^{22}	62
$fr \rightarrow en$	0	19.8	50.0	0k	30 / 1	17
	2	21.5	53.6	20k	36 / 18	27
	4	21.9	57.5	32k	73 / 10^7	43
	∞	21.7	56.6	50k	10^3 / 10^{27}	59
$en \rightarrow de$	0	12.3	37.8	0k	27 / 1	17
	2	12.5	38.6	64k	31 / 2	18
	4	12.7	42.8	87k	65 / 10^5	26
	∞	12.7	45.9	102k	10^2 / 10^{22}	33
$de \rightarrow en$	0	17.0	43.0	0k	28 / 1	16
	2	17.5	44.8	71k	34 / 3	19
	4	18.1	49.0	92k	68 / 10^5	26
	∞	18.0	50.0	105k	10^2 / 10^{26}	33
$en \rightarrow cs$	0	9.9	30.1	0k	27 / 1	27
	2	9.9	30.5	33k	29 / 1	27
	4	9.9	33.1	46k	55 / 10^3	34
	∞	9.9	34.8	57k	10^2 / 10^{21}	47
$cs \rightarrow en$	0	14.7	36.2	0k	23 / 1	29
	2	14.6	36.5	30k	25 / 1	29
	4	14.8	40.2	41k	52 / 10^4	39
	∞	14.9	44.0	51k	10^2 / 10^{21}	51

Tableau 7.2 – Impact de la stratégie de filtrage des règles (*max_coût*), en utilisant le jeu d'étiquettes de PdD, sur le corpus de test : scores BLEU obtenues par NCODE et par le décodage oracle ; nombre de règles de réordonnancement (#règles) ; taille des treillis de réordonnancement (nombre moyen d'arcs / nombre moyen de chemins) ; et couverture (voir 6.9).

observe que si le nombre de règles est réduit de moitié pour $en \rightarrow de$ par rapport à $en \rightarrow fr$, la taille des espaces de réordonnancement est comparable, malgré une couverture bien moindre pour $en \rightarrow de$. La paire de langues anglais-tchèque se retrouve avec les espaces de réordonnancement les plus petits, mais sans que leur couverture en soit affectée pour autant. De manière intéressante, cette dernière paire de langues, que l'on considère complexe du point de vue des réordonnancements généralement impliqués, présente quasiment le double de traductions monotones par rapport aux autres paires de langues [8], malgré des scores BLEU significativement moindres, que ce soit avec NCODE ou en décodage oracle, ce qui indique à nouveau que les réordonnancements ne peuvent expliquer qu'en partie la complexité de cette paire de langues.

En filtrant moins les règles de réordonnancement, on observe des espaces de réordonnancement de taille plus importante, avec une meilleure couverture et des scores BLEU oracles plus élevés. Pour la paire de langues anglais-français, dans les conditions standard de test, nous observons cependant que cette augmentation de la taille de l'espace de recherche s'accompagne d'une légère baisse du score BLEU. Ceci montre l'importance du compromis lorsque l'on construit l'espace de réordonnancement, car des réordonnancements rares ou de mauvaise qualité peuvent introduire des alternatives plausibles, mais fallacieuses. On n'observe cependant pas ce phénomène pour les autres paires de langues, pour lesquelles le score BLEU ne change pas vraiment lorsque la taille des treillis de réordonnancements augmente, et avec elle le pourcentage de cas où le réordonnancement de référence est présent dans l'espace de recherche. En particulier, la direction de traduction $en \rightarrow cs$ n'est ni pénalisée ni avantagée lorsque l'on considère d'autres réordonnancements possibles que la permutation monotone. Nous pensons que ceci est, encore une fois, un indicateur montrant que l'enjeu principal pour améliorer nos systèmes de traduction ne se trouve pas uniquement dans le design de l'espace de réordonnancement.

Il peut sembler surprenant, à première vue, que les règles de faible coût ne changent pratiquement pas les espaces de réordonnancement engendrés. Par exemple, pour $en \rightarrow de$, les $64k$ règles avec un coût d'au plus 2 conduisent à des espaces de réordonnancement qui ne contiennent en moyenne que deux chemins. De manière analogue, $33k$ règles pour $en \rightarrow cs$ n'engendrent en général aucune autre permutation que le chemin monotone. Ceci montre que la plupart des règles extraites ne se généralisent pas bien au delà du corpus d'apprentissage, car la séquence d'étiquettes correspondante est trop rare pour être observée sur le corpus de test. Ces règles ne sont pas filtrées par un critère qui est conditionnel, car elles correspondent à des séquences d'étiquettes rares. Cependant, même si elle sont inefficaces, elles ne sont pas non plus nuisibles, puisqu'elles ne s'appliquent presque jamais.

8. Ce qui explique d'ailleurs en grande partie la bonne couverture.

7.6.6 Différents jeux d'étiquettes

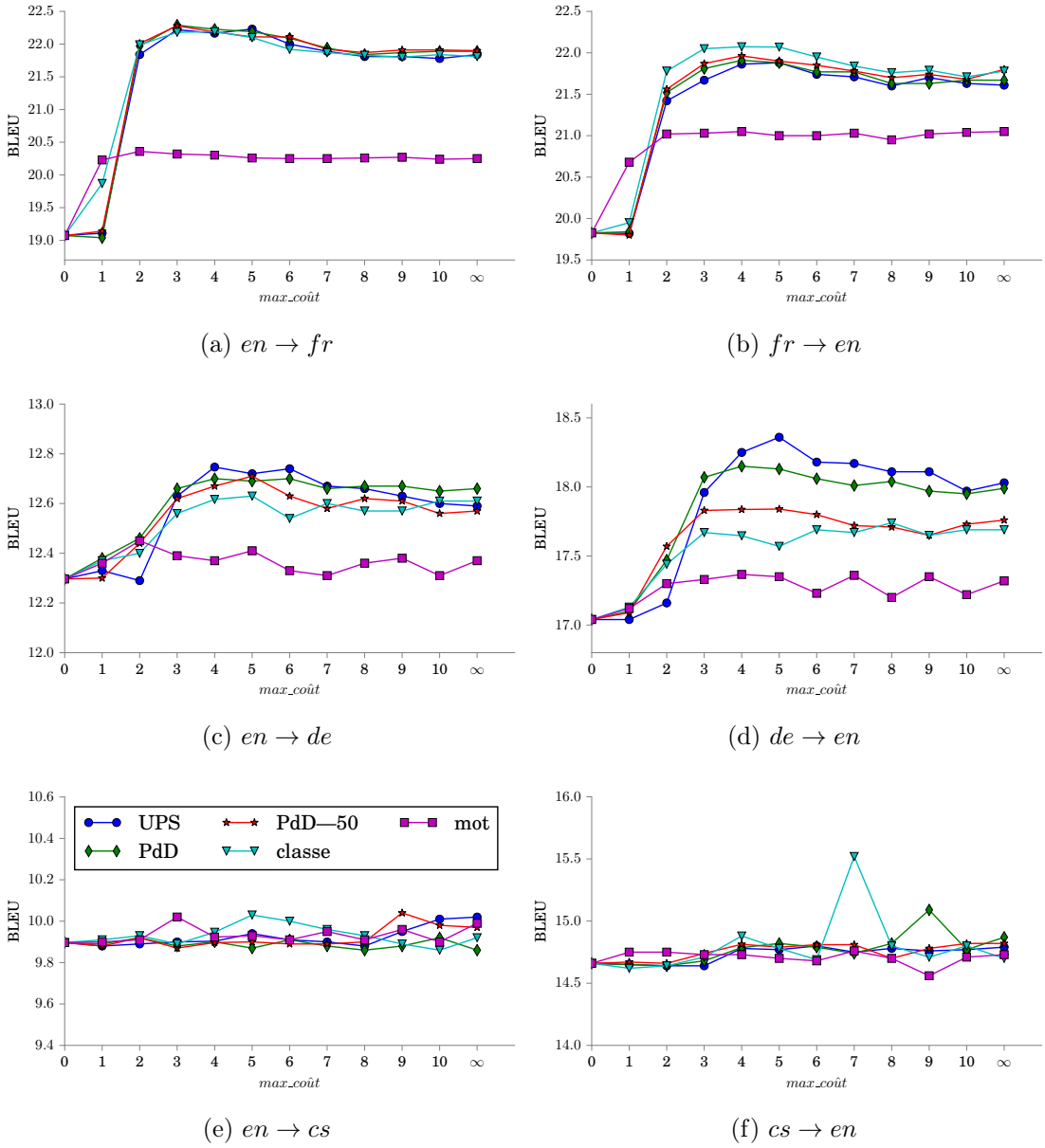
La figure 7.3 donne les résultats obtenus lorsque les règles de réordonnancement sont construites sur différents jeux d'étiquettes et différents seuils de filtrage. Il est intéressant de remarquer que si les comportements diffèrent suivant les paires de langues, ils sont relativement semblables au sein d'une même paire pour les deux directions de traduction. À la fois pour anglais-français et anglais-allemand, l'utilisation de règles entièrement lexicalisées, conduit aux plus mauvais résultats, avec quelques gains cependant par rapport au cas monotone. Pour la paire de langues anglais-français, nous relevons peu de différences selon les autres jeux d'étiquettes, mais observons que les résultats se dégradent légèrement lorsque le paramètre de filtrage augmente au delà d'un certain seuil. Il est cependant quelque peu surprenant de voir que pour $fr \rightarrow en$, les règles construites à partir des classes de mots obtiennent des résultats légèrement supérieurs aux autres jeux d'étiquettes, ce qui était difficile à prévoir au vu de la figure 6.4 (a) du chapitre précédent. Les classes de mots se comportent cependant moins avantageusement pour la paire de langues anglais-allemand, pour laquelle nous constatons d'ailleurs des différences plus prononcées entre les jeux d'étiquettes. Pour cette paire de langues, les jeux d'étiquettes plus petits semblent préférables, probablement en raison de la plus grande généralisation qu'ils permettent.

Comme précédemment, nous observons un comportement différent pour la paire de langues anglais-tchèque. Pour $en \rightarrow cs$, nous ne notons d'ailleurs aucun changement important lorsque les paramètres varient. Ce résultat permet cependant de réfuter un argument qui voudrait que la direction $cs \rightarrow en$ soit pénalisée par un jeu d'étiquettes plus grand que les autres paires de langue, puisque l'utilisation du jeu réduit d'étiquettes universelles conduit aux mêmes résultats.

De manière générale, des résultats au moins aussi bons sont obtenus avec les jeux d'étiquettes à grain grossier ou avec les classes de mots, ce qui montre que l'on peut, sans perte, utiliser ces derniers, ce qui est particulièrement intéressant pour des langues peu dotées en ressources ou en outils linguistiques.

7.6.7 Comparaison avec MJ-i

Le tableau 7.3 propose une comparaison face à face entre les contraintes MJ-i et celles de notre approche à base de règles. Les espaces de réordonnancement donnés par les contraintes MJ dépassent de plusieurs ordres de grandeur en taille ceux des règles de réordonnancement, mais n'amènent qu'à des performances égales ou inférieures. Ceci justifie l'utilisation de règles syntaxiques, plutôt que le recours à toutes les permutations locales possibles et corrobore à nouveau le compromis discuté plus haut. Les règles de réordonnancement permettent à la fois de diminuer significativement la taille de l'espace de réordonnancement— et donc d'accélérer



(a) $en \rightarrow fr$ (b) $fr \rightarrow en$

(c) $en \rightarrow de$ (d) $de \rightarrow en$

(e) $en \rightarrow cs$ (f) $cs \rightarrow en$

FIGURE 7.3 – Comparaison des performances (score BLEU, un seul calibrage de MIRA) pour différents jeux d'étiquettes sur les corpus de test en fonction du seuil de filtrage ($max\_coût$).

	$en \rightarrow fr$		$fr \rightarrow en$		$en \rightarrow de$		$de \rightarrow en$		$en \rightarrow cs$		$cs \rightarrow en$	
	BLEU	taille	BLEU	taille	BLEU	taille	BLEU	taille	BLEU	taille	BLEU	taille
max_long=2	21.8	10^2	21.4	10^2	12.5	77	17.2	7	9.9	7	14.7	67
MJ-1	21.5	10^{14}	21.3	10^{17}	12.5	10^{14}	17.2	10^{19}	9.9	10^{14}	14.7	10^{14}
max_long=3	22.1	10^4	21.7	10^5	12.6	10^3	17.5	10^2	9.9	10^2	15.0	10^3
MJ-2	21.7	10^{24}	21.5	10^{28}	12.5	10^{24}	17.3	10^{31}	9.9	10^{24}	15.0	10^{24}
max_long=4	22.3	10^4	21.9	10^6	12.6	10^4	17.7	10^4	9.9	10^3	14.8	10^4
MJ-3	21.7	10^{30}	21.6	10^{36}	12.5	10^{30}	17.5	10^{40}	9.9	10^{30}	14.8	10^{30}

Tableau 7.3 – Comparaison entre l'approche à base de règles en fonction de la taille limite autorisée (max_long) et des contraintes purement combinatoires (MJ-i). Les scores BLEU sont des moyennes entre trois calibrages indépendants avec MIRA.

le décodage, sans affecter ni sa couverture, ni les performances du système global. À titre d'exemple, pour $en \rightarrow fr$, l'étape de calibrage avec MJ-3 prend à peu près vingt fois plus de temps qu'avec $maxlen = 4$.

7.6.8 Discussion

L'un des travaux les plus proches du nôtre est certainement celui de Herrmann *et al.* (2013b), dans lequel les auteurs ont indépendamment exploré des réordonnancements oracles pour analyser le potentiel d'une approche de préordonnancement et l'impact de différentes contraintes de réordonnancement. Leurs résultats, pour la paire de langues anglais-allemand qu'ils ont étudiée, sont similaires aux nôtres, bien que notre étude ne rejoigne que partiellement leurs conclusions. Si la différence entre le meilleur réordonnancement et le réordonnancement de référence montre effectivement que des améliorations sont à attendre du côté du design des espaces de recherche, nous pensons notre condition entre *règle* et *aug* plus informative. En effet, l'absence d'amélioration significative lorsque l'on ajoute le réordonnancement de référence dans le treillis de règles montre que seuls de faibles gains sont à attendre si l'on ne s'intéresse qu'à l'espace de réordonnancement sans changer les modèles qui les évaluent.

7.7 Conclusions

Dans le présent chapitre, nous avons comparé des espaces de recherche de réordonnancement engendrés par différentes règles de réordonnancement ainsi que par des contraintes de permutations locales. Les règles de réordonnancement basées sur des informations linguistiques ou pseudo-linguistiques permettent de construire des espaces beaucoup plus compacts tout en améliorant les performances de traduction, et ne dépendent que de manière marginale des facteurs de mots utilisés pour augmenter leur puissance de généralisation. Pour étudier le potentiel d'un meilleur espace de réordonnancement nous avons utilisé notre système de traduction n-gramme, qui permet de séparer les étapes de réordonnancement et de décodage, mais nous pensons que les enseignements tirés de cette étude ont une portée plus générale, en particulier pour les systèmes qui permettent d'encoder l'espace de réordonnancement dans un treillis avant le décodage de traduction. Le cadre choisi nous a permis d'étudier l'importance et l'impact du choix de l'espace de réordonnancement sur les performances du système global. Si nous observons qu'il y a encore du chemin à faire pour améliorer les espaces de réordonnancement explorés, la marge de gain semble cependant assez faible. Ceci est d'autant plus vrai pour des paires de langues comme anglais-tchèque, la plus difficile des paires de langues considérées ici, ce qui suggère que le design de l'espace de réordon-

nancement n'est pas, à l'heure actuelle, la source principale de problème dans le design du système global. Ainsi, des améliorations de l'espace de réordonnancement devront s'accompagner d'une amélioration dans les modèles d'évaluation des hypothèses, et en particulier dans les modèles de réordonnancement, si l'on veut pouvoir observer des gains intéressants.

Il est cependant important de comprendre que dans ce travail, nous nous sommes attachés à étudier l'importance et l'expressivité de *l'espace de recherche de réordonnancement* au moment du décodage, toutes choses étant égales par ailleurs. Les divergences dans l'ordre des mots entre les langues interviennent à bien d'autres niveaux que simplement dans l'espace de recherche (alignement, définition des unités, définition des segments/tuples, etc.). L'importance de l'espace de réordonnancement dans notre travail doit être rapportée au fait que nous utilisons une approche (particulière) à base de segments qui dépend d'un certain alignement, fixé ici une fois pour toutes. Ce que nous croyons, c'est que dans l'état actuel de nos systèmes, et du moins pour les langues étudiées ici, la principale source d'erreur — même du point de vue des réordonnancements — ne semble pas être dans le design de l'espace de réordonnancement. Ceci ne signifie cependant pas que l'ordre des mots ne joue pas un rôle critique dans le processus entier de traduction. En particulier, il conditionne l'extraction de tuples qui fonde tout le modèle à base de segments que nous utilisons. Enfin, les erreurs d'alignements sont aussi en partie imputables à des divergences dans l'ordre des mots. Elles interviennent à de nombreux niveaux dans le cadre que nous étudions, affectant à la fois les performances des systèmes et les analyses que nous en faisons. Il serait alors également important d'étudier et de mieux comprendre l'impact des erreurs d'alignement sur les réordonnancements à proprement parler.

Enfin, nous ne nous sommes pas directement attachés à évaluer la qualité de l'ordre des mots dans les traductions proposées mais plutôt la qualité globale des traductions, en étudiant les phénomènes de réordonnancement. Il est également intéressant d'étudier l'impact de l'espace choisi sur la qualité des réordonnancements obtenus, en utilisant par exemple d'autres métriques que BLEU, plus sensibles à la composante de réordonnancement. Dans (Pécheux *et al.*, 2016a), nous avons utilisé LRscore (Birch et Osborne, 2010) et BEER (Stanojević et Sima'an, 2014), mais les effets observés et les conclusions obtenues restent sensiblement les mêmes que pour BLEU.

Chapitre 8

Conclusions

Sommaire

Dans ce dernier chapitre de conclusion générale, nous rappelons brièvement les questions principales à l'origine de ce travail et résumons l'ensemble des contributions présentées dans ce manuscrit qui permettent d'y apporter des éléments de réponse. Si les cas d'études que nous avons menés permettent de mieux comprendre les enjeux de la définition des espaces de recherche ou de supervision pour différentes tâches de TAL, le design d'un espace de recherche optimal dans le cas général reste un problème entier. Nous présentons donc un ensemble de perspectives de suites possibles ouvertes par nos travaux ainsi que quelques nouvelles questions soulevées par les expériences menées au cours de la thèse.

8.1 Sur l'importance de l'espace de recherche en traitement automatique des langues

Il existe de nombreuses raisons de s'intéresser à l'espace de recherche dans les problèmes d'apprentissage automatique et en particulier en TAL, comme nous l'avons discuté en introduction et au chapitre 2. L'une des principales motivations est de savoir réduire efficacement la taille de ces espaces et ainsi diminuer considérablement la complexité des problèmes — du moins du point de vue du temps de traitement. Pour de nombreux problèmes de TAL, comme celui posé par la tra-

duction automatique présenté au chapitre 5, la réduction du nombre d'hypothèses possibles est même impérative.

Cela nous a conduit à étudier l'utilisation de contraintes pour réduire l'espace de recherche, dans deux configurations : (1) l'utilisation de contraintes de types, qui peuvent par exemple être issues de dictionnaires, étude que nous avons menée dans le chapitre 4 ; et (2) l'utilisation de règles de réordonnancement pour limiter les permutations envisagées en traduction automatique, ce qui fait l'objet du chapitre 6.

En dehors du gain apporté par une exploration plus rapide, une autre question légitime consiste à se demander si des contraintes sur l'espace de recherche, qui peuvent traduire une connaissance extérieure, permettent ou non d'améliorer les performances, questionnement qui a été le sujet du chapitre 4. Nous avons vu que l'intégration telle quelle de contraintes lors de l'apprentissage pouvait, au contraire, dégrader les résultats et qu'il était nécessaire de conserver une forme de contraste dans l'espace de recherche.

Un espace de recherche plus précis, contenant moins d'hypothèses invraisemblables, pourrait également permettre d'améliorer les performances, notamment en limitant les erreurs de recherche. Nous avons réalisé cette étude dans le chapitre 7 pour le problème des réordonnancements en traduction automatique.

Pour améliorer les méthodes de TAL, il faut, d'une part, chercher à obtenir de meilleurs modèles — pour attribuer le meilleur score à l'hypothèse recherchée, et, d'autre part, proposer un espace de recherche adapté. À titre de diagnostic, pour orienter au mieux de futures recherches, il est intéressant d'étudier le potentiel relatif de l'amélioration de chacune de ces deux pistes. Dans le chapitre 7, nous avons étudié cet équilibre dans le cas de l'espace de réordonnancement en traduction automatique.

Enfin, le chapitre 3 s'intéresse à une question un peu différente, mais proche de celle de l'espace de recherche : celle de l'espace de référence. Les contraintes de types, ainsi que d'autres contraintes provenant du transfert d'annotations à partir d'une autre langue, peuvent également être utilisées pour définir un espace de référence, cependant ambigu et non plus formé d'une seule hypothèse de référence comme c'est le cas dans le cadre usuel. Il s'agit d'un cadre d'apprentissage automatique moins favorable, néanmoins nécessaire lorsque l'on s'intéresse à des langues peu dotées en ressources. Dans le chapitre 3, nous avons ainsi étudié les conditions pour lesquelles l'apprentissage avec de tels espaces est possible et proposé une nouvelle méthode dans ce cadre ambigu.

Ainsi, dans ce travail, nous nous sommes intéressés à l'importance et à l'impact des espaces de recherche et de supervision, en adoptant plusieurs points de vue et en menant des études de cas pour diverses tâches de traitement automatique des langues. Ce travail a ainsi permis d'apporter quelques nouveaux éléments de

réponse à des questions importantes en TAL concernant les espaces de recherche et de supervision :

- Quel est le meilleur espace de recherche ?
- Comment le constuire ?
- Est-il avantageux de restreindre l'espace de recherche aux hypothèses les plus prometteuses et dans quelle mesure ?
- Dans les modèles actuels, le design de l'espace de recherche est-il limitant ?
- Peut-on également apprendre lorsque l'on dispose d'un espace de supervision ambigu ?

8.2 Contributions

Nous détaillons par la suite quelques éléments de réponse que nous avons pu apporter à ces questions en soulignant les contributions de ce travail.

Espace de recherche lors de l'apprentissage, identification d'un paradoxe En s'intéressant à l'introduction de contraintes de types lors de l'apprentissage d'un étiqueteur morpho-syntaxique nous avons identifié un paradoxe : l'utilisation d'un espace de recherche réduit, permettant effectivement d'améliorer les résultats lors du décodage, dégrade cependant les performances globales lorsqu'il est utilisé à l'apprentissage. L'intuition qu'un espace de recherche réduit et précis, même parfaitement, est nécessairement avantageux n'est donc pas toujours correcte.

Étude sur les contraintes de types dans les modèles CRF Nous avons conduit une étude détaillée sur l'utilisation de contraintes de types pour réduire l'espace de recherche des CRF. Nos résultats expérimentaux montrent que l'utilisation de contraintes entraîne une forme de sous-apprentissage, en ignorant la contribution de nombreux exemples et en réduisant ainsi la capacité de généralisation. Dans le cas de l'analyse morpho-syntaxique étudié, cela concerne principalement les mots hors-vocabulaire, mais la portée de nos observations nous semble plus générale. Ainsi, un espace de recherche restreint lors de l'apprentissage est susceptible de nuire à la généralisation, même lorsque les mêmes contraintes seront appliquées par la suite.

Conserver une forme de constraste dans l'espace de recherche lors de l'apprentissage Nos expériences nous ont conduit à conclure qu'il semble important de limiter, lors de l'apprentissage, l'impact des contraintes afin de garder une forme de contraste. Ainsi un « bon » espace de recherche, est également un

espace de recherche qui permet, lors de l'apprentissage, un contraste suffisant entre les exemples de références et les hypothèses négatives. Il n'y a donc pas nécessairement un seul espace de recherche pour un problème donné mais il est possible, voire avantageux, d'utiliser un espace de recherche différent lors de l'apprentissage — plus large pour permettre davantage de constraste – et lors du décodage — plus restreint et plus précis.

Tranfert cross-lingue vu comme un problème d'apprentissage ambigu Nous avons étudié le problème de l'apprentissage d'un analyseur morpho-syntaxique lorsque les étiquettes de supervision ne sont que partiellement connues, par exemple lorsque celles-ci sont automatiquement transférées à partir d'une langue source plus riche en annotations et avons formulé ce problème dans le cadre de l'apprentissage ambigu (Bordes *et al.*, 2010; Cour *et al.*, 2011). Les contraintes utilisées pour restreindre l'espace de recherche peuvent également être utilisées pour définir un espace de supervision ambigu lorsque que cela est nécessaire, ce qui nous a conduit à étudier ce cadre.

Un nouveau modèle à base d'historique pour l'apprentissage ambigu En abordant le problème du transfert cross-lingue sous l'angle de l'apprentissage ambigu, nous avons montré qu'il était possible d'étendre un modèle à base d'historique en adaptant une mise à jour de type perceptron au contexte faiblement supervisé. Ce modèle obtient des performances comparables aux modèles précédemment proposés, avec, pour certaines langues, des résultats qui surpassent l'état de l'art dans le domaine.

Peut-on apprendre avec un espace de supervision ambigu ? Nous avons mené une série d'expériences contrôlées et dans un cadre pratique pour comprendre le comportement d'un modèle CRF partiellement observé et du modèle à base d'historique. Celles-ci permettent de montrer que l'apprentissage avec un espace de supervision ambigu est possible, sous certaines conditions sur la quantité et la variété de l'ambiguïté. En particulier, une proportion faible mais suffisante des occurrences doit être entièrement désambiguïsée.

Comprendre le rôle des ressources et des contraintes Nous avons comparé l'utilisation de différentes ressources et contraintes pour comprendre leur importance dans le cadre étudié. Les contraintes de types jouent un rôle important, à la fois pour obtenir un filtrage efficace des contraintes d'occurrences, pour compléter celles-ci et, lors du test, pour réduire les étiquettes possibles. Cela nécessite cependant de disposer de contraintes de types avec une couverture suffisamment importante.

Nous avons montré qu'à condition d'utiliser au mieux les ressources pendant l'apprentissage et l'inférence, les deux méthodes étudiées dépassent les meilleurs résultats publiés jusqu'ici pour dix langues de familles différentes, dans certains cas de manière importante. Notre analyse montre même qu'il y a probablement plus à gagner en se concentrant sur de meilleures ressources plutôt que sur les méthodes d'apprentissage à proprement parler. Les travaux de Lacroix *et al.* (2016b) renforcent ces conclusions.

Les limites du transfert cross-lingue pour les langues peu dotées dans la pratique Nous avons montré que la combinaison de données parallèles et monolingues était nécessaire pour permettre des résultats satisfaisants. Cette observation rend finalement incertaine l'applicabilité de ces méthodes dans un véritable contexte de langues peu dotées, ou lorsque les langues source et cible ne sont pas suffisamment proches pour que le transfert de PdD fasse sens et que les liens d'alignement soient sûrs, problème repris par Aufrant *et al.* (2016).

Si le cadre de transfert cross-lingue se justifie naturellement par le manque de données annotées pour des langues peu dotées, il ne semble pas évident que toutes les conditions que nous avons identifiées pour avoir de bons résultats soient effectivement rencontrées dans un véritable scénario.

Analyse critique de l'évaluation de la tâche de transfert cross-lingue Nous avons conduit une analyse critique de la tâche de transfert cross-lingue pour l'apprentissage d'un analyseur morpho-syntaxique dans son ensemble ainsi qu'une analyse approfondie des erreurs des modèles étudiés. Ceci nous a permis de soulever les difficultés et les limites que pose l'évaluation de telles méthodes, avec en particulier les problèmes liés aux conventions d'annotation et les divergences de domaines.

Règles de réordonnancement comme contraintes pour les espaces de recherche en traduction automatique Nous avons mené une étude du système à base de règles syntaxiques de Ncode qui justifie son utilisation et montre ses limites. L'étude empirique des réordonnancements nous a permis de les répartir en trois grandes familles en fonction de leur taille, et suivant laquelle l'approche à base de règle fonctionne plus ou moins bien. L'approche à base de règles syntaxiques en utilisant des facteurs de mots est efficace pour les réordonnancements de faible portée. Pour les réordonnancements de moyenne portée, il est également utile de filtrer les permutations possibles. En revanche, les plus grands déplacements ne peuvent que peu ou pas être pris en compte par cette approche.

Importance du design de l'espace de réordonnancement Nous avons proposé une étude complète sur le compromis à faire lors du choix de l'espace de réordonnancement. Nous avons étudié l'importance du design de cet espace, en utilisant un système de traduction état de l'art que nous avons contribué à développer. En représentant tous les réordonnancements dans un treillis de permutations avant le décodage, nous avons pu étudier directement l'espace de recherche et ainsi mesurer son impact sur le processus complet de traduction. Cela nous a permis également de comparer directement diverses manières de construire cet espace et d'étudier ses différentes propriétés. Des expériences oracles nous ont permis de comprendre le potentiel éventuel des différents espaces dans le meilleur des cas, ainsi que l'influence des erreurs de recherche et des erreurs de modèles sur la qualité finale. Nous avons mené les expériences pour trois paires de langues qui diffèrent du point de vue des réordonnancements à modéliser.

Nos résultats expérimentaux montrent que les erreurs de recherche et les erreurs de modèles semblent avoir un impact plus important que le design de l'espace de réordonnancement. Il nous semble donc qu'il est plus important de se concentrer en premier lieu sur la manière dont est évalué l'ordre des mots plutôt que sur l'expressivité de l'espace de recherche. Ainsi des améliorations de l'espace de recherche doivent s'accompagner d'améliorations des modèles de réordonnancement, si l'on veut pouvoir observer des gains intéressants.

8.3 Perspectives

Les études que nous avons menées dans ce travail ont apporté de nouveaux éclairages sur le rôle de l'espace de recherche, en particulier pour l'analyse morpho-syntaxique et les réordonnancements en traduction automatique. Mais il reste encore de nombreuses questions, dont de nouvelles soulevées par nos analyses et expériences. La fin de ce chapitre est consacrée à quelques pistes de développement possibles pour poursuivre le travail amorcé.

Contraintes de types pour restreindre l'espace de recherche Les solutions simples que nous avons proposées pour restreindre l'espace avec des contraintes de types tout en conservant une forme de contraste restent insatisfaisantes et ne permettent pas de retrouver les performances maximales atteintes en l'absence de contraintes à l'apprentissage dans le cas général. Nous pensons cependant qu'il est possible d'exploiter ce type de contraintes en trouvant des solutions plus adaptées. Une idée qui semble prometteuse est de simuler formellement le fait que d'autres étiquettes sont possibles, en présence de contraintes, sans avoir à effecteur les calculs, coûteux, lors de l'apprentissage. Pour cela, il faudrait réussir à intégrer dans

le modèle les compétiteurs à une étiquette en tant que groupe et non considérer toutes les étiquettes indépendamment.

Vers des espaces de recherche adaptés pour l'apprentissage Nous avons identifié qu'une forme de contraste était nécessaire dans les espaces de recherche pour que l'apprentissage soit effectif. En soi, ceci n'est pas surprenant. Dans le cas extrême, si l'espace d'apprentissage est réduit à une seule alternative — tel un questionnaire à choix multiples qui ne comporterait qu'une seule possibilité, on comprend qu'aucun apprentissage n'est possible. Ce que nous avons en revanche mis en lumière, et qui n'apparaissait pas au premier abord, c'est que dans des configurations qui ne sont jamais observées, il y a une forme d'information implicite qui permet une meilleure généralisation. Il serait ainsi intéressant de comprendre quelles contraintes ne retirent aucune information et lesquelles encodent des configurations impossibles mais néanmoins utiles à un meilleur apprentissage.

Intégrer des informations linguistiques lors de l'apprentissage Nous avons proposé d'intégrer les contraintes de types dans l'espace de recherche comme source possible d'information linguistique externe (lorsque le dictionnaire est issu d'une base de connaissance externe, par exemple Wiktionnaire). Nous pensons que notre étude illustre également les cas où l'on utilise d'autres manières d'intégrer une information, par exemple des contraintes sur des séquences d'étiquettes possibles.

D'autres manières d'intégrer les informations linguistiques dans les modèles sont possibles. Pour les contraintes de types par exemple, il est courant d'utiliser une caractéristique supplémentaire qui indique si tel mot est présent dans un dictionnaire ou si telle association d'un mot et d'une étiquette est autorisée par les contraintes (Müller *et al.*, 2013). Cette intégration dans les caractéristiques permet au modèle de calibrer l'importance relative de ces contraintes, dites « douces », par opposition aux contraintes directement dans l'espace de recherche, dites « dures ». Cependant, cette approche ne permet pas de réduire l'espace de recherche, question qui était au centre de nos propos. La meilleure manière d'intégrer des informations linguistiques externes reste donc une problématique intéressante à étudier.

Une première piste serait d'utiliser le cadre de la régularisation *a posteriori* (Ganchev *et al.*, 2010) ou celui des modèles conditionnels sous contraintes (Chang *et al.*, 2010, 2012). Une autre idée qui nous semble prometteuse serait d'adapter le cadre de l'estimation contrastive (Smith et Eisner, 2005) au cas discriminant et d'exploiter les informations linguistiques disponibles pour construire le meilleur espace de constraste possible. Enfin, les modèles de Müller *et al.* (2013) devraient pouvoir être étendus à d'autres informations linguistiques, en enchaînant des modèles de

complexité croissante dans lesquels les connaissances linguistiques pourraient être intégrées au fur et à mesure.

Évaluation extrinsèque des résultats d'analyse morpho-syntaxique par transfert cross-lingue Si les résultats que nous avons obtenus dans le cadre du transfert cross-lingue pour la tâche d'analyse morpho-syntaxique restent en deçà des résultats du cadre supervisé, malgré une évaluation sans doute pessimiste, il nous semble que les performances sont proches des meilleurs résultats possibles, compte tenu des ressources disponibles. Toutefois la tâche d'analyse morpho-syntaxique est rarement un but en soi, mais plutôt un prétraitement utilisé en TAL pour des tâches plus complexes, comme l'analyse en dépendance ou la traduction automatique. Nous prévoyons donc d'effectuer une évaluation de la qualité des analyseurs ainsi appris sur les performances globales de tâches plus générales. Seule cette analyse extrinsèque pourra déterminer si les différences entre le cadre faiblement supervisé et le cadre supervisé ont une importance en pratique. Enfin, une évaluation indirecte, sur une tâche concrète, permettrait également de considérer de véritables langues peu dotées — que l'absence de corpus annotés ne nous a pas permis d'inclure dans notre analyse.

Traitement multi-lingue et langues peu-dotées Avec la popularisation des usages du réseau Internet dans le monde et l'avènement des objets intelligents, le besoin de nouvelles méthodes de TAL ne cessera de grandir et le marché de s'étendre à l'international, en particulier pour de nouvelles langues peu couvertes aujourd'hui. L'intérêt pour les langues peu dotées en ressources et les méthodes associées est donc amené à se généraliser, ainsi que la création de nouvelles ressources multi-lingues, comme le projet *Universal Dependencies*[1].

L'une des perspectives naturelles de nos travaux dans ce domaine est d'explorer de nouvelles tâches de TAL. Lacroix *et al.* (2016a) ont appliqué des idées similaires à celles que nous avons présentées dans le cadre de l'analyse en dépendance. D'autres travaux se sont également intéressés à d'autres tâches comme la reconnaissance d'entités nommées (Wang et Manning, 2014b). La traduction automatique, dans une configuration où l'on disposerait de plusieurs fragments redondants et partiellement traduits, pourrait être une autre application du cadre d'apprentissage ambigu en TAL.

Enfin, dans le contexte du TAL multi-lingue, les réseaux de neurones et l'apprentissage profond pourront jouer un rôle important, par exemple pour projeter les annotations (Zennaki *et al.*, 2015), en apprenant implicitement des éléments de représentations universelles (Gillick *et al.*, 2015) ou en permettant de transférer

1. http://universaldependencies.org

les représentations ou les modèles appris entre les langues (Kozhevnikov et Titov, 2014; Zhang *et al.*, 2016).

Des règles de réordonnancement plus adaptées Notre étude sur les règles de réordonnancement a montré que les permutations observées dans les corpus d'apprentissage se généralisaient plus ou moins bien à celles de test en fonction de leur taille. Nous avons donc suggéré d'adapter la granularité des règles en fonction de la taille des réordonnancements. On pourrait par exemple utiliser des classes de mots hiérarchiques (Ushioda, 1996) et considérer des facteurs plus ou moins hauts dans la hiérarchie en fonction de la taille des règles. Une perspective intéressante est donc de mettre en œuvre cette extension du système de réordonnancement à base de règles et d'évaluer son efficacité.

Comparer d'autres manières possibles de génération des treillis de réordonnancement Nous avons établi une méthodologie qui permet de comparer les espaces de recherche encodés sous forme de treillis, ou, en généralisant, de forêts. Nous avons comparé des treillis de permutations engendrés par des règles syntaxiques ou en imposant des contraintes sur les permutations. De nombreuses autres méthodes pour créer les espaces de recherche existent ; on peut par exemple adapter la plupart des méthodes de préordonnancement pour obtenir un treillis et non une simple hypothèse. Il serait ainsi intéressant de compléter notre étude avec des treillis de réordonnancement obtenus par diverses autres techniques de préordonnancement, par exemple, celles de (Tromble et Eisner, 2009), (Visweswariah *et al.*, 2011) ou plus récemment de (Stanojević et Sima'an, 2015). On pourrait également utiliser d'autres systèmes de traduction, par exemple Moses [2], qui permet également un décodage monotone avec un treillis comme entrée.

Envisager de nouvelles manières de construire l'espace de réordonnancements Les travaux de préordonnancement visent le plus souvent à produire un unique et meilleur réordonnancement de la source. Il serait intéressant de les adapter vers l'objectif particulier d'un meilleur espace de recherche et d'optimiser les paramètres de ces modèles en fonction de la qualité des espaces de recherche ainsi obtenus. Nous pensons que l'étude menée dans ce travail constitue une première étape vers cet objectif.

Distinguer les erreurs de recherche et les erreurs de modèle Nous avons montré que les erreurs de modèle étaient, dans notre étude, le premier facteur d'amélioration potentiel. En effet, même dans un espace de recherche composé

2. http://www.statmt.org/moses/

de deux réordonnancements possibles, le système de traduction se trompait à de nombreuses reprises. Nous n'avons cependant pas évalué précisément l'impact de la taille de l'espace de réordonnancement sur les erreurs de recherche, ce qui nous permettrait de compléter notre étude.

Vers un modèle discriminant unifié pour la traduction automatique Dans le chapitre 5 nous avons remarqué que l'architecture globale des systèmes de traduction, en dehors de quelques développements récents, était sensiblement la même et pouvait se décomposer en deux grandes étapes. Une première étape de collecte de statistiques, généralement effectuée sur de très grands corpus de données parallèles ou monolingues, et une deuxième phase de calibrage des différents sous-modèles ainsi constitués, généralement coûteuse et effectuée sur de petits corpus du domaine d'intérêt.

La plupart des développements récents se sont principalement attelés à améliorer la deuxième étape, de manière à la rendre plus robuste, plus rapide ou de manière à y intégrer directement un grand nombre de caractéristiques, dites « creuses ». Intégrer une myriade de nouvelles caractéristiques riches dans les modèles existants a été, et est encore, l'objet de nombreux travaux (Liang *et al.*, 2006; Chiang *et al.*, 2008, 2009; Cherry, 2013)

Réécrivons l'équation (5.7) de façon à expliciter *tous* les paramètres qui interviennent dans la formulation du modèle complet :

$$p_{\boldsymbol{\Theta},\boldsymbol{\Gamma}}(\mathbf{y},\mathbf{d}|\mathbf{x}) = \frac{1}{\mathcal{Z}_{\boldsymbol{\Theta},\boldsymbol{\Gamma}}(\mathbf{x})} \exp\left(\sum_{k=1}^{K} \theta_k \phi_k(\mathbf{x},\mathbf{y},\mathbf{d},\boldsymbol{\gamma}_k)\right) \tag{8.1}$$

où pour $k \in [\![1, K]\!]$, $\phi_k(\mathbf{x},\mathbf{y},\mathbf{d},\boldsymbol{\gamma}_k)$ est l'un des sous-modèles, paramétrée par $\boldsymbol{\gamma}_k$ et $\mathcal{Z}_{\boldsymbol{\Theta},\boldsymbol{\Gamma}}(\mathbf{x})$ la fonction de partition. On a regroupé l'ensemble des paramètres des différents sous-modèles dans $\boldsymbol{\Gamma} = (\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K)$ ainsi que l'ensemble des poids $\boldsymbol{\Theta} = (\theta_1, \ldots, \theta_K)$.

L'approche classique que nous avons vue au chapitre 5, introduite par Och et Ney (2002) et reprise par la suite dans la grande majorité des travaux, comporte deux étapes. La première consiste à apprendre les paramètres $\boldsymbol{\gamma}_k$ de chaque sous-modèle sur le corpus d'apprentissage de manière indépendante. La deuxième étape est de calibrer, sur le corpus de développement, les poids θ_k de ces modèles, en y ajoutant éventuellement de nouvelles caractéristiques creuses.

Pourtant, ce processus en deux étapes présente plusieurs inconvénients :

- Les poids associés aux différents modèles sont typiquement estimés sur des corpus dits de développement, de taille très modeste. Ceci pose un problème important de surapprentissage, en particulier lorsque l'on souhaite inclure

un nombre important de modèles[3], et *a fortiori* si l'on souhaite inclure des millions de caractéristiques supplémentaires.

- Cette architecture semble conduire à une estimation sous-optimale des paramètres puisque les paramètres des modèles sont estimés suivant une procédure en deux étapes, dont la première ignore tout de la deuxième. Les paramètres γ_k des sous-modèles sont estimés indépendamment de la tâche finale et ne tiennent pas compte de la manière dont sera évalué le modèle complet.
- La grande majorité des approches discriminantes pour la traduction se concentrent sur un petit nombre d'hypothèses de traduction : les M-meilleures hypothèses données, ce qui risque d'introduire des biais et ne tient pas compte des erreurs de recherche (Liu et Huang, 2014).
- Ce processus en deux étapes ne permet pas d'intégrer simplement de nouvelles informations ou des connaissances externes aux corpus d'entraînement. Par exemple, il n'est pas aisé d'introduire des lexiques bilingues, que l'on peut par exemple extraire de WIKTIONNAIRE, de corpus comparables ou de dictionnaires de termes médicaux. Comme ceux-ci ne constituent pas des corpus de même nature que les corpus de phrases, il n'est pas forcément aisé d'estimer les modèles lors de la première phase et, sans les scores associés, on ne voit pas très bien comment les utiliser lors de la phase de calibrage.

Pour toutes ces raisons il serait intéressant de disposer d'un cadre entièrement discriminant, dans lequel tous les paramètres seraient appris de manière unifiée, discriminante et en une seule fois. À notre connaissance, peu de travaux ont abordé le problème de la traduction automatique sous cet angle, c'est-à-dire en abordant directement l'optimisation de (8.1). Tillmann et Zhang (2006) considèrent un modèle de séquences de blocs, un cadre relativement différent des modèles à base de segments, mais dont tous les poids sont appris de manière discriminante, directement sur le corpus d'apprentissage. (Arun et Koehn, 2007) utilisent un jeu de caractéristiques qui correspond à l'union de ceux des sous-modèles, et sans utiliser ces derniers. Dans (Blunsom *et al.*, 2008), pour un modèle hiérarchique à base de segments, les seules caractéristiques considérées sont des indicateurs binaires sur l'identité des règles. Ce travail a été poursuivi dans (Blunsom et Osborne, 2008) de manière à pouvoir y intégrer un modèle de langue. Lavergne *et al.* (2011) abordent la traduction automatique comme un problème d'apprentissage structuré et apprennent tous les poids avec un modèle CRF, mais leur approche nécessite la connaissance de la segmentation et du réordonnancement pour obtenir de bons résultats. Cette idée est développée dans (Lavergne *et al.*, 2013b) dans un modèle entièrement unifié comme celui de Blunsom *et al.* (2008).

3. La méthode la plus utilisée, MERT, devient prohibitivement instable dès que l'on dépasse une dizaine de modèles (Hopkins et May, 2011)

Nous avons commencé à poursuivre les travaux de Lavergne *et al.* (2013b) et proposé une architecture dans laquelle le processus de traduction est entièrement modélisé sous la forme d'un problème d'apprentissage structuré, en utilisant la famille des modèles exponentiels donnée par l'équation (1.1) comme modèle paramétrique. Ce système est, à notre connaissance, le seul à travailler directement sur l'espace de recherche complet, sans approximations. La question de l'espace de réordonnancement occupe donc une place importante puisque des problèmes similaires à ceux observés dans le chapitre 4 sont susceptibles d'avoir un impact conséquent. Nous avons comparé cette nouvelle approche avec le système état de l'art NCODE sur une tâche de traduction relativement simple pour la paire de langue français-anglais, dans les deux sens. Si les gains obtenus sont modérés, il est intéressant de noter qu'un tel système, encore en développement, se compare déjà favorablement à un système déjà bien établi. En revanche, lorsque nous abordons des corpus plus conséquents et d'autres paires de langues comme anglais-allemand ou anglais-tchèque, les performances de notre modèle sont encore en deçà de celles du système état de l'art. De nombreux facteurs peuvent expliquer les difficultés rencontrées par ce nouveau cadre et nous nous sommes donc attachés à en comprendre les raisons, en considérant des sous-problèmes plus simples et en proposant des solutions pour les résoudre. Nous avons par exemple introduit de nouvelles fonctions objectifs pour l'optimisation et une version « douce » de la fonction de perte *hope-and-fear* (Gimpel et Smith, 2010) nous a permis d'obtenir des résultats très encourageants. L'achèvement de ces travaux devra nous permettre de disposer d'un formalisme bien fondé pour intégrer les multiples sources d'information qui sont nécessaires à la prédiction d'une traduction correcte ; de ce point de vue, cette piste de travail reste d'actualité.

Réseaux de neurones et apprentissage profond Depuis quelques années, les modèles de réseaux de neurones, les représentations distribuées et les méthodes d'apprentissage profond ont connu des développements importants et des résultats très compétitifs, qui ont peu à peu nourri tous les champs d'applications du TAL. L'un des intérêts de ces approches est de ne plus dépendre de caractéristiques sélectionnées manuellement, qui demandent des connaissances linguistiques ou une forme d'expertise dans la tâche en question, mais d'intégrer implicitement le choix des caractéristiques au sein du modèle.

On peut remarquer en particulier de nouveaux modèles de traduction par réseaux de neurones profonds (Cho *et al.*, 2014; Sutskever *et al.*, 2014), dont l'approche se distingue des modèles à deux étages présentés au chapitre 5. Ces modèles utilisent des réseaux de neurones récurrents — similaires à ceux utilisés pour les modèles de langue — pour encoder une phrase source par une représentation vectorielle. Cette représentation est ensuite utilisée pour initialiser un réseau récurrent

qui permet de générer la traduction. Diverses extensions ont également vu le jour, en particulier l'utilisation de modèles d'attention (Luong *et al.*, 2015). Comme pour notre modèle à base d'historique (§ 3.4.3), il est préférable lors de l'apprentissage de ne pas utiliser la séquence de référence mais d'entraîner le modèle de façon à lui permettre de corriger ses propres erreurs (Bengio *et al.*, 2015).

Les questions liées à l'espace de recherche se posent différemment pour ces modèles. Schématiquement, le problème de la recherche de la meilleure hypothèse dans l'espace de recherche est remplacé par une étape de génération. Il est toujours possible d'utiliser le modèle comme une fonction de score utilisée pour identifier la meilleure hypothèse dans un espace de recherche. Mais il est également possible d'utiliser directement le réseau récurrent pour *générer* une sortie, qui n'est peut-être pas l'hypothèse optimale, mais dont la génération ne dépend pas directement de la taille de l'espace des hypothèses possibles. Remarquons cependant que cette étape de génération peut poser des problèmes semblables à ceux que nous avons rencontrés dans le cas des espaces de recherche. Lors de l'apprentissage d'un système de traduction, le vocabulaire de sortie peut être de taille importante, et la somme sur tous les mots de ce vocabulaire, nécessaire pour calculer la log-vraisemblance, est alors coûteuse. On utilise souvent dans ce cas une approximation par échantillonnage, mais on pourrait imaginer introduire ici des contraintes semblables à des contraintes de types. Notons que de nombreux travaux proposent même de modéliser les problèmes de TAL directement à partir des caractères et non des mots, par exemple (Zhang *et al.*, 2015) pour la classification de textes, (Kim *et al.*, 2015) pour des modèles de langages neuronaux ou (Ling *et al.*, 2015; Chung *et al.*, 2016) pour la traduction — voire même directement à partir de l'encodage du texte en octet (Gillick *et al.*, 2015). Il serait intéressant de comprendre exactement quels sont les espaces de recherches explorés, implicitement, par ces nouveaux modèles.

Dans un monde de plus en plus connecté, où le numérique prend peu à peu place au sein de tous les objets de notre entourage, nous pensons que le traitement automatique des langues et les méthodes associées joueront un rôle très important dans l'intégration des objets intelligents dans notre quotidien. Il semble vraisemblable que l'interaction avec les machines prendra peu à peu la forme de dialogues avec des agents conversationnels, avec anticipation de nos questions et prédiction de nos réponses. Si la puissance de calcul ne cesse et ne cessera encore d'augmenter, nous pensons que le développement de modèles toujours plus complexes sur de nouveaux supports ne fera pas disparaître les questions d'ordre computationnel et donc l'intérêt pour les espace de recherche.

Publications de l'auteur

Articles publiés dans des revues scientifiques avec comité de lecture

Nicolas Pécheux, Alexandre Allauzen, Jan Niehues et François Yvon : Reordering Space Design in Statistical Machine Translation. *Language Resources and Evaluation Journal*, 50(2):375–410, 2016a. ISSN 1574-0218. URL http://dx.doi.org/10.1007/s10579-016-9353-8. *[Cité pages 23, 147, 171, et 192]*

Nicolas Pécheux, Guillaume Wisniewski et François Yvon : Reassessing the Value of Resources for Cross-Lingual Transfer of POS Tagging Models. *Language Resources and Evaluation Journal*, 2016b. *[Cité pages 23, 58, et 93]*

Articles publiés dans des actes de conférences avec comité de lecture

Hervé Bredin, Anindya Roy, Nicolas Pécheux et Alexandre Allauzen : "Sheldon Speaking, Bonjour !" : Leveraging Multilingual Tracks for (Weakly) Supervised Speaker Identification. *In Proceedings of the ACM International Conference on Multimedia*, MM'14, pages 137–146. Association for Computer Machinery, 2014. URL http://doi.acm.org/10.1145/2647868.2654929. *[Aucune citation]*

Nicolas Pécheux, Alexandre Allauzen, Thomas Lavergne, Guillaume Wisniewski et François Yvon : Oublier ce qu'on sait, pour mieux apprendre ce qu'on ne sait pas : une étude sur les contraintes de type dans les modèles CRF. *In Conférence sur le Traitement Automatique des Langues Naturelles*, TALN'15. Association pour le Traitement Automatique des Langues, 2015. URL http://www.atala.org/taln_archives/TALN/TALN-2015/taln-2015-long-004.pdf. *[Cité pages 23 et 104]*

Nicolas Pécheux, Alexandre Allauzen et François Yvon : Rule-based Reordering Space in Statistical Machine Translation. *In Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC'14, pages 1800–1806. European Language Resources Association, 2014. URL http://aclanthology.info/papers/rule-based-reordering-space-in-statistical-machine-translation.
[Cité pages 23, 147, et 171]

Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham et François Yvon : Cross-Lingual Part-of-Speech Tagging through Ambiguous Learning. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP'14, pages 1779–1785. Association for Computational Linguistics, 2014a. URL http://aclanthology.info/papers/cross-lingual-part-of-speech-tagging-through-ambiguous-learning.
[Cité pages 23, 58, 74, et 118]

Guillaume Wisniewski, Nicolas Pécheux, Elena Knyazeva, Alexandre Allauzen et François Yvon : Apprentissage partiellement supervisé d'un étiqueteur morpho-syntaxique par transfert cross-lingue. *In Conférence sur le Traitement Automatique des Langues Naturelles*, TALN'14, pages 173–183. Association pour le Traitement Automatique des Langues, 2014b. URL http://aclweb.org/anthology/F14-1016. *[Cité pages 23, 58, et 74]*

Articles publiés dans des actes de workshops avec comité de lecture

Alexandre Allauzen, Nicolas Pécheux, Khanh Quoc Do, Marco Dinarelli, Thomas Lavergne, Aurélien Max, Hai-Son Le et François Yvon : LIMSI @ WMT13. *In Proceedings of the Eighth Workshop on Statistical Machine Translation*, WMT'13, pages 62–69. Association for Computational Linguistics, 2013.
[Cité pages 140, 160, et 161]

Benjamin Marie, Alexandre Allauzen, Franck Burlot, Quoc-Khanh Do, Julia Ive, elena knyazeva, Matthieu Labeau, Thomas Lavergne, Kevin Löser, Nicolas Pécheux et François Yvon : LIMSI@WMT'15 : Translation Task. *In Proceedings of the Tenth Workshop on Statistical Machine Translation*, WMT'15, pages 145–151. Association for Computational Linguistics, 2015. URL http://aclanthology.info/papers/limsi-wmt-15-translation-task. *[Cité page 140]*

Nicolas Pécheux, Li Gong, Khanh Quoc Do, Benjamin Marie, Yulia Ivanishcheva, Alexander Allauzen, Thomas Lavergne, Jan Niehues, Aurélien Max et François Yvon : LIMSI @ WMT'14 Medical Translation Task. *In Proceedings of the Ninth Workshop on Statistical Machine Translation*, WMT'14, pages 246–253. Association for Computational Linguistics, 2014. URL http://aclanthology.info/papers/limsi-wmt-14-medical-translation-task. *[Cité page 140]*

Guillaume Wisniewski, Nicolas Pécheux, Alexander Allauzen et François Yvon : LIMSI Submission for WMT'14 QE Task. *In Proceedings of the Ninth Workshop on Statistical Machine Translation*, WMT'14, pages 348–354. Association for Computational Linguistics, 2014. URL http://aclanthology.info/papers/limsi-submission-for-wmt-14-qe-task. *[Cité page 140]*

Guillaume Wisniewski, Nicolas Pécheux et François Yvon : Why Predicting Post-Edition is so Hard ? Failure Analysis of LIMSI Submission to the APE Shared Task. *In Proceedings of the Tenth Workshop on Statistical Machine Translation*, WMT'15, pages 222–227. Association for Computational Linguistics, 2015. URL http://aclweb.org/anthology/W15-3027. *[Cité page 140]*

Bibliographie

Anne Abeillé, Lionel Clément et François Toussenel : Building a treebank for French. *In* Anne Abeillé, éditeur : *Treebanks : Building and Using Parsed Corpora*. Kluwer, Dordrecht, 2003. *[Cité page 61]*

Alexandre Allauzen, Josep M. Crego, İlknur Durgar El-Kahlout et François Yvon : LIMSI's Statistical Translation Systems for WMT'10. *In Proceedings of the Joint Workshop on Statistical Machine Translation and Metrics*, pages 54–59, Uppsala, Sweden, 2010. *[Cité page 160]*

Alexandre Allauzen et François Yvon : *Méthodes statistiques pour la traduction automatique*, chapitre 7, pages 271–356. Hermès, Paris, 2011. *[Cité page 127]*

Yasemin Altun, Mark Johnson et Thomas Hofmann : Investigating Loss Functions and Optimization Methods for Discriminative Learning of Label Sequences. *In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 145–152, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1119355.1119374. *[Cité pages 36, 42, et 55]*

Massih-Reza Amini : *Apprentissage machine de la théorie à la pratique*. Algorithms. Eyrolles, May 2015. *[Cité page 31]*

Galen Andrew et Jianfeng Gao : Scalable Training of L1-regularized Log-linear Models. *In Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 33–40, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. URL http://doi.acm.org/10.1145/1273496.1273501. *[Cité pages 51 et 52]*

Abhishek Arun et Philipp Koehn : Online learning methods for discriminative training of phrase based statistical machine translation. *In Proceedings of MT Summit XI*, volume 2, 2007. *[Cité page 203]*

Lauriane Aufrant, Guillaume Wisniewski et François Yvon : Cross-lingual alignment transfer : a chicken-and-egg story ? *In Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP*, pages 35–44, San Diego, California, June 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W16-1205. *[Cité page 197]*

Michael Auli, Michel Galley et Jianfeng Gao : Large-scale Expected BLEU Training of Phrase-based Reordering Models. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1250–1260, Doha, Qatar, October 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1132. *[Cité page 173]*

Michael Auli, Adam Lopez, Hieu Hoang et Philipp Koehn : A Systematic Analysis of Translation Model Search Spaces. *In Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 224–232, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1626431.1626475. *[Cité pages 21, 129, et 174]*

Satanjeev Banerjee et Alon Lavie : METEOR : An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. *In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. *[Cité pages 138, 159, et 177]*

Michele Banko et Robert C. Moore : Part of Speech Tagging in Context. *In Proceedings of the 20th International Conference on Computational Linguistics*, COLING'04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. *[Cité page 62]*

David Bellos : *Le poisson et le bananier. Une histoire fabuleuse de la traduction.* Flammarion, Paris, 2012. *[Cité pages 16 et 134]*

Samy Bengio, Oriol Vinyals, Navdeep Jaitly et Noam M. Shazeer : Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. *In Advances in Neural Information Processing Systems*, 2015. URL http://arxiv.org/abs/1506.03099. *[Cité page 205]*

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero et Dan Klein : Painless Unsupervised Learning with Features. *In Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, NAACL HLT'10, pages 582–590, Los Angeles, California, June 2010. *[Cité pages 62 et 64]*

Adam L. Berger, Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Andrew S. Kehler et Robert L. Mercer : Language Translation Apparatus and Method Using Context-based Translation Models, 1996. *[Cité pages 137, 152, et 153]*

Alexandra Birch : *Reordering Metrics for Statistical Machine Translation.* Thèse de doctorat, University of Edinburgh, 2011. *[Cité page 153]*

Alexandra Birch et Miles Osborne : LRscore for Evaluating Lexical and Reordering Quality in MT. *In Proceedings of the Joint Workshop on Statistical Machine*

Translation and Metrics, pages 327–332, Uppsala, Sweden, July 2010. Association for Computational Linguistics. *[Cité page 192]*

Alexandra Birch, Miles Osborne et Phil Blunsom : Metrics for MT Evaluation : Evaluating Reordering. *Machine Translation*, 24(1):15–26, mars 2010. *[Cité page 177]*

Alexandra Birch, Miles Osborne et Philipp Koehn : Predicting Success in Machine Translation. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, Honolulu, USA, 2008. Association for Computational Linguistics. *[Cité page 148]*

Arianna Bisazza et Marcello Federico : Dynamically Shaping the Reordering Search Space of Phrase-Based Statistical Machine Translation. *Transactions of the Association of Computational Linguistics – Volume 1*, pages 327–340, 2013a. *[Cité page 174]*

Arianna Bisazza et Marcello Federico : Efficient Solutions for Word Reordering in German-English Phrase-Based Statistical Machine Translation. *In Proceedings of the Workshop on Statistical Machine Translation*, WMT, pages 440–451, Sofia, Bulgaria, August 2013b. Association for Computational Linguistics. *[Cité page 174]*

Christopher M. Bishop : *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738. *[Cité page 28]*

Ezra Black, Fred Jelinek, John Lafferty, David M. Magerman, Robert Mercer et Salim Roukos : Towards History-based Grammars : Using Richer Models for Probabilistic Parsing. *In Proceedings of the Workshop on Speech and Natural Language*, HLT'91, pages 134–139, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics. *[Cité page 79]*

Phil Blunsom, Trevor Cohn et Miles Osborne : A Discriminative Latent Variable Model for Statistical Machine Translation. *In Proceedings of ACL-08 : HLT*, pages 200–208, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P08/P08-1024. *[Cité pages 31 et 203]*

Phil Blunsom et Miles Osborne : Probabilistic Inference for Machine Translation. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 215–223, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1613715.1613746. *[Cité page 203]*

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia et Aleš Tamchyna : Findings of the 2014 Workshop on Statistical Machine Translation. *In Proceedings of the Workshop on Statistical Machine Translation*, WMT, pages 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. *[Cité page 140]*

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut et Lucia Specia : Findings of the 2013 Workshop on Statistical Machine Translation. *In Proceedings of the Workshop on Statistical Machine Translation*, WMT, pages 1–44, Sofia, Bulgaria, 2013. *[Cité page 140]*

Antoine Bordes, Nicolas Usunier et Jason Weston : Label Ranking under Ambiguous Supervision for Learning Semantic Correspondences. *In Proceedings of the International Conference on Machine Learning*, ICML'10, pages 103–110, 2010. *[Cité pages 59, 64, 75, 76, 77, 81, 82, 88, 89, et 196]*

Guillaume Bouchard : Bias-variance tradeoff in hybrid generative-discriminative models. *In The Sixth International Conference on Machine Learning and Applications*, ICMLA, pages 124–129, 2007. URL http://dx.doi.org/10.1109/ICMLA.2007.85. *[Cité page 36]*

Guillaume Bouchard et Bill Triggs : The Trade-off Between Generative and Discriminative Classifiers. *In* Jaromír Antoch, éditeur : *16th IASC International Symposium on Computational Statistics, COMPSTAT 2004, August, 2004*, pages 697–704, Prague, Tchéquie, 2004. Physica-Verlag/Springer. *[Cité page 36]*

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith et Hans Uszkoreit : TIGER : Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2(4):597–620, 2004. ISSN 1570-7075. *[Cité page 109]*

Jürgen Broschart : Why Tongan does it differently : Categorial distinctions in a language without nouns and verbs. *Linguistic Typology*, 1:123–166, 10 2009. *[Cité page 65]*

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra et Jenifer C. Lai : Class-based N-gram Models of Natural Language. *Comput. Linguist.*, 18(4):467–479, décembre 1992. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=176313.176316. *[Cité pages 62, 132, et 156]*

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra et Robert L. Mercer : The Mathematics of Statistical Machine Translation : Parameter Estimation. *Computational Linguistics*, 19(2):263–311, juin 1993. URL http://dl.acm.org/citation.cfm?id=972470.972474. *[Cité pages 134 et 136]*

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut et Lucia Specia : Findings of the 2012 Workshop on Statistical Machine Translation. *In Proceedings of the Workshop on Statistical Machine Translation*, WMT, pages 10–51, Montréal, Canada, 2012. *[Cité pages 140 et 160]*

Marine Carpuat, Yuval Marton et Nizar Habash : Improving Arabic-to-English statistical machine translation by reordering post-verbal subjects for alignment. *In Proceedings of the Annual Meeting on Association for Computational Linguistics*, ACL, pages 178–183, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. *[Cité page 150]*

Francesco Casacuberta et Enrique Vidal : Machine Translation with Inferred Stochastic Finite-State transducers. *Computational Linguistics*, 30(3):205–225, 2004. *[Cité page 141]*

Ming-Wei Chang, Dan Goldwasser, Dan Roth et Vivek Srikumar : Discriminative Learning over Constrained Latent Representations. *In The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 429–437, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. *[Cité pages 122 et 199]*

Ming-Wei Chang, Lev Ratinov et Dan Roth : Structured Learning with Constrained Conditional Models. *Machine Learning*, 88(3):399–431, juin 2012. *[Cité pages 122 et 199]*

Eugene Charniak et Mark Johnson : Coarse-to-fine N-best Parsing and MaxEnt Discriminative Reranking. *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 173–180, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1219840.1219862. *[Cité page 122]*

Stanley F. Chen et Joshua T. Goodman : An Empirical Study of Smoothing Techniques for Language Modeling. Rapport technique TR-10-98, Computer Science Group, Harvard University, 1998. *[Cité page 121]*

Colin Cherry : Improved Reordering for Phrase-Based Translation using Sparse Features. *In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 22–31, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N13-1003. *[Cité pages 173 et 202]*

Colin Cherry, Robert C. Moore et Chris Quirk : On Hierarchical Re-ordering and Permutation Parsing for Phrase-based Decoding. *In Proceedings of the Workshop on Statistical Machine Translation*, WMT, pages 200–209, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. *[Cité pages 138 et 152]*

David Chiang : A Hierarchical Phrase-based Model for Statistical Machine Translation. *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL, pages 263–270. Association for Computational Linguistics, 2005. *[Cité page 133]*

David Chiang : Hope and Fear for Discriminative Training of Statistical Translation Models. *J. Mach. Learn. Res.*, 13(1):1159–1187, avril 2012. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2503308.2343684. *[Cité page 50]*

David Chiang, Kevin Knight et Wei Wang : 11,001 New Features for Statistical Machine Translation. *In Proceedings of Human Language Technologies : The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 218–226, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-41-1. URL http://dl.acm.org/citation.cfm?id=1620754.1620786. *[Cité page 202]*

David Chiang, Yuval Marton et Philip Resnik : Online Large-margin Training of Syntactic and Structural Translation Features. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 224–233, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1613715.1613747. *[Cité page 202]*

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)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1179. *[Cité page 204]*

Christos Christodoulopoulos, Sharon Goldwater et Mark Steedman : Two Decades of Unsupervised POS Induction : How Far Have We Come ? *In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 575–584, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. *[Cité page 62]*

Junyoung Chung, Kyunghyun ; Cho et Yoshua Bengio : A Character-Level Decoder without Explicit Segmentation for Neural Machine Translation. *ArXiv e-prints*, mars 2016. *[Cité page 205]*

Jonathan H. Clark, Chris Dyer, Alon Lavie et Noah A. Smith : Better Hypothesis Testing for Statistical Machine Translation : Controlling for Optimizer Instability. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, HLT, pages 176–181. Association for Computational Linguistics, 2011. *[Cité page 162]*

Shay B. Cohen, Dipanjan Das et Noah A. Smith : Unsupervised Structure Prediction with Non-Parallel Multilingual Guidance. *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, EMNLP'11, pages 50–61, Edinburgh, Scotland, UK., 2011. *[Cité page 63]*

Michael Collins : Discriminative Training Methods for Hidden Markov Models : Theory and Experiments with Perceptron Algorithms. *In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP, pages 1–8, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1118693.1118694. *[Cité page 49]*

Michael Collins : Head-Driven Statistical Models for Natural Language Parsing. *Computational Linguistics*, 29(4):589–637, décembre 2003. *[Cité page 79]*

Michael Collins, Philipp Koehn et Ivona Kucerova : Clause Restructuring for Statistical Machine Translation. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics*, ACL, pages 531–540, Ann Arbor, Michigan, 2005. *[Cité pages 149 et 150]*

Marta Ruiz Costa-jussà et José A. R. Fonollosa : Statistical Machine Reordering. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 70–76, Sydney, Australia, July 2006. Association for Computational Linguistics. *[Cité page 151]*

Maria R. Costa-Jussà et J. A. R. Fonollosa : State-of-the-art word reordering approaches in statistical machine translation : A survey. *IEICE Transactions on Information and Systems 92*, (11):2179–2185, 2009. *[Cité page 137]*

Timothee Cour, Ben Sapp et Ben Taskar : Learning from Partial Labels. *Journal of Machine Learning Research*, 12:1501–1536, juillet 2011. *[Cité pages 59, 64, 75, 76, 77, 81, 82, 88, 89, et 196]*

Josep M. Crego : *Architecture and Modeling for N-gram-based Statistical Machine Translation*. Thèse de doctorat, Universitat Politècnica de Catalunya, 2008. *[Cité page 157]*

Josep M. Crego, Marta R. Costa-jussà, José B. Mariño et José A. R. Fonollosa : Ngram-based versus Phrasebased Statistical Machine Translation. *In Proceedings of International Workshop on Spoken Language Translation*, IWSLT, pages 177–184, 2005. *[Cité page 142]*

Josep M. Crego et Nizar Habash : Using Shallow Syntax Information to Improve Word Alignment and Reordering for SMT. *In Proceedings of the Workshop on Statistical Machine Translation*, WMT, pages 53–61, Columbus, Ohio, June 2008. Association for Computational Linguistics. *[Cité page 150]*

Josep M. Crego et José B. Mariño : Improving Statistical MT by Coupling Reordering and Decoding. *Machine Translation*, 20(3):199–215, 2006. *[Cité pages 141, 143, 152, et 155]*

Josep M. Crego, François Yvon et José B. Mariño : N-code : an open-source Bilingual N-gram SMT Toolkit. *Prague Bulletin of Mathematical Linguistics*, 96:49–58, 2011. *[Cité page 141]*

Fabien Cromières : *Vers un plus grand lien entre alignement, segmentation et structure des phrases*. Thèse de doctorat, Université de Grenoble, France, 2010. *[Cité page 136]*

Dipanjan Das et Slav Petrov : Unsupervised Part-of-speech Tagging with Bilingual Graph-based Projections. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, HLT '11, pages 600–609, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. *[Cité pages 63, 67, et 88]*

Hal Daumé, III et Daniel Marcu : Learning As Search Optimization : Approximate Large Margin Methods for Structured Prediction. *In Proceedings of the 22nd International Conference on Machine Learning*, ICML'05, pages 169–176, New York, NY, USA, 2005. ACM. *[Cité pages 79 et 81]*

Hal Daumé, III et Daniel Marcu : Domain Adaptation for Statistical Classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126, mai 2006. URL http://dl.acm.org/citation.cfm?id=1622559.1622562. *[Cité page 97]*

Adrià de Gispert, Marcus Tomalin et Bill Byrne : Word Ordering with Phrase-Based Grammars. *In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 259–268, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E14-1028. *[Cité page 148]*

Adrià de Gispert et José B. Mariño : Linguistic Tuple Segmentation in N-gram-Based Statistical Machine Translation. *In Proceedings of the European Conference on Speech Communication and Technology*, INTERSPEECH. ISCA, 2006. *[Cité page 143]*

Daniel Déchelotte, Gilles Adda, Alexandre Allauzen, Olivier Galibert, Jean-Luc Gauvain, Hélène Maynard et François Yvon : LIMSI's statistical translation systems for WMT'08. *In Proceedings of the NAACL-HTL Statistical Machine Translation Workshop*, Columbus, Ohio, 2008. *[Cité page 160]*

John DeNero et Jakob Uszkoreit : Inducing Sentence Structure from Parallel Corpora for Reordering. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 193–203, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. *[Cité page 151]*

Michael Deza et Tayuan Huang : Metrics on Permutations, a Survey. *Journal of Combinatorics, Information and System Sciences*, 1998. *[Cité page 176]*

Bonnie J. Dorr : Machine Translation Divergences : a formal description and proposed solution. *Computational Linguistics*, 20(4):597–633, 1994. *[Cité page 68]*

Mark Dredze, Partha Pratim Talukdar et Koby Crammer : Sequence Learning from Data with Multiple Labels. *In Proceedings of the ECML-PKDD 2009 Workshop on Learning from Multi-Label Data*, MLD, 2009. *[Cité pages 76, 77, et 88]*

Markus Dreyer, Keith B. Hall et Sanjeev P. Khudanpur : Comparing Reordering Constraints for SMT Using Efficient BLEU Oracle Computation. *In Proceedings of th Workshop on Syntax and Structure in Statistical Translation*, NAACL-HLT, pages 103–110, Rochester, New York, 2007. *[Cité pages 152 et 178]*

Long Duong, Trevor Cohn, Karin Verspoor, Steven Bird et Paul Cook : What Can We Get From 1000 Tokens ? A Case Study of Multilingual POS Tagging For Resource-Poor Languages. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP'14, pages 886–897, Doha, Qatar, October 2014a. Association for Computational Linguistics. *[Cité pages 61, 95, 97, 98, et 101]*

Long Duong, Trevor Cohn, Karin Verspoor, Steven Bird et Paul Cook : What Can We Get From 1000 Tokens ? A Case Study of Multilingual POS Tagging For Resource-Poor Languages. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 886–897, Doha, Qatar, October 2014b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1096. *[Cité page 95]*

Ilknur Durgar El-Kahlout et François Yvon : The pay-offs of preprocessing for German-English Statistical Machine Translation. *In* Marcello Federico, Ian Lane, Michael Paul et François Yvon, éditeurs : *Proceedings of International Workshop on Spoken Language Translation*, IWSLT, pages 251–258, 2010. *[Cité page 160]*

Greg Durrett, Adam Pauls et Dan Klein : Syntactic Transfer Using a Bilingual Lexicon. *In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL'12, pages 1–11, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. *[Cité page 63]*

Chris Dyer et Philip Resnik : Context-free Reordering, Finite-state Translation. *In Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 858–866, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL http://dl.acm.org/citation.cfm?id=1857999.1858127. *[Cité pages 150 et 158]*

Bradley Efron : The Efficiency of Logistic Regression Compared to Normal Discriminant Analysis. *Journal of the American Statistical Association*, 70(352):892–898, 1975. URL http://www.jstor.org/stable/2285453. *[Cité page 36]*

Jakob Elming et Nizar Habash : Syntactic reordering for English-Arabic phrase-based machine translation. *Proceedings of the EACL Workshop on Computational Approaches to Semitic Languages*, page 69, 2009. *[Cité page 150]*

Nicholas Evans et Stephen C. Levinson : The myth of language universals : Language diversity and its importance for cognitive science. *Behavioral and Brain Sciences*, 32:429–448, 10 2009. *[Cité page 65]*

Yang Feng, Haitao Mi, Yang Liu et Qun Liu : An Efficient Shift-reduce Decoding Algorithm for Phrased-based Machine Translation. *In Proceedings of the 23rd International Conference on Computational Linguistics : Posters*, COLING, pages 285–293, Stroudsburg, PA, USA, 2010a. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1944566.1944599. *[Cité page 137]*

Yang Feng, Haitao Mi, Yang Liu et Qun Liu : An Efficient Shift-reduce Decoding Algorithm for Phrased-based Machine Translation. *In Proceedings of the International Conference on Computational Linguistics*, COLING, pages 285–293, Stroudsburg, PA, USA, 2010b. Association for Computational Linguistics. *[Cité page 152]*

Alexander Fraser, Helmut Schmid, Richárd Farkas, Renjing Wang et Hinrich Schütze : Knowledge Sources for Constituent Parsing of German, a Morphologically Rich and Less-Configurational Language. *Volume 39, Issue 1 - March 2013*, 2013. *[Cité page 109]*

Souhir Gahbiche-Braham, Hélène Bonneau-Maynard, Thomas Lavergne et François Yvon : Joint Segmentation and POS Tagging for Arabic Using a CRF-based Classifier. *In Proceedings of the Language Resources and Evaluation Conference*, LREC '12, Istanbul, Turkey, 2012. *[Cité page 104]*

Michel Galley, Mark Hopkins, Kevin Knight et Daniel Marcu : What's in a translation rule ? *In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, HLT-NAACL, 2004. *[Cité pages 133 et 152]*

Michel Galley et Christopher D. Manning : A Simple and Effective Hierarchical Phrase Reordering Model. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 848–856, Stroudsburg, PA, USA, 2008a. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1613715.1613824. *[Cité page 137]*

Michel Galley et Christopher D. Manning : A Simple and Effective Hierarchical Phrase Reordering Model. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 848–856. Association for Computational Linguistics, 2008b. *[Cité page 149]*

Kuzman Ganchev et Dipanjan Das : Cross-Lingual Discriminative Learning of Sequence Models with Posterior Regularization. *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP'13, pages 1996–2006, Seattle, Washington, USA, October 2013a. *[Cité page 64]*

Kuzman Ganchev et Dipanjan Das : Cross-Lingual Discriminative Learning of Sequence Models with Posterior Regularization. *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1996–2006, Seattle, Washington, USA, October 2013b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1205. *[Cité page 95]*

Kuzman Ganchev, Jennifer Gillenwater et Ben Taskar : Dependency Grammar Induction via Bitext Projection Constraints. *In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL '09, pages 369–377, Stroudsburg, PA, USA, 2009. *[Cité page 63]*

Kuzman Ganchev, João Graça, Jennifer Gillenwater et Ben Taskar : Posterior Regularization for Structured Latent Variable Models. *J. Mach. Learn. Res.*, 11:2001–2049, août 2010. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1756006.1859918. *[Cité pages 64, 122, et 199]*

Qin Gao et Stephan Vogel : Parallel implementations of word alignment tool. *In Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57, 2008. *[Cité page 67]*

Dan Garrette et Jason Baldridge : Learning a Part-of-Speech Tagger from Two Hours of Annotation. *In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, NAACL'13, pages 138–147, Atlanta, Georgia, June 2013. Association for Computational Linguistics. *[Cité page 61]*

Dmitriy Genzel : Automatically Learning Source-side Reordering Rules for Large Scale Machine Translation. *In Proceedings of the International Conference on Computational Linguistics*, COLING, pages 376–384. Association for Computational Linguistics, 2010. *[Cité pages 149 et 150]*

D. Gillick, C. Brunk, O. Vinyals et A. Subramanya : Multilingual Language Processing From Bytes. *ArXiv e-prints*, novembre 2015. *[Cité pages 200 et 205]*

Kevin Gimpel et Mohit Bansal : Weakly-Supervised Learning with Cost-Augmented Contrastive Estimation. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1329–1341, Doha, Qatar, October 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1139. *[Cité page 122]*

Kevin Gimpel et Noah A. Smith : Softmax-margin CRFs : Training Log-linear Models with Cost Functions. *In Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 733–736, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL http://dl.acm.org/citation.cfm?id=1857999.1858111. *[Cité pages 50 et 204]*

Kevin Gimpel et Noah A. Smith : Structured Ramp Loss Minimization for Machine Translation. *In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, NAACL HLT '12, pages 221–231, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. ISBN 978-1-937284-20-6. URL http://dl.acm.org/citation.cfm?id=2382029.2382059. *[Cité pages 50 et 51]*

Chooi-Ling Goh, Takashi Onishi et Eiichiro Sumita : Rule-based Reordering Constraints for Phrase-based SMT. *In Proceedings of the Conference of the European Association for Machine Translation*, pages 113–120, 2011. *[Cité page 152]*

Edward Goldberg : FoG : Synthesizing forecast text directly from weather maps. *In Proceedings of the Ninth Conference on Artificial Intelligence for Applications*, pages 156–162, 1993. *[Cité page 128]*

Yoav Goldberg, Meni Adler et Michael Elhadad : EM Can Find Pretty Good HMM POS-Taggers (When Given a Good Start). *In Proceedings of ACL-08 : HLT*, pages 746–754. Association for Computational Linguistics, 2008. URL http://aclweb.org/anthology/P08-1085. *[Cité page 121]*

Jan Hajic : Morphological Tagging : Data vs. Dictionaries. *In 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000. *[Cité page 121]*

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue et Yi Zhang : The CoNLL-2009 Shared Task : Syntactic and Semantic Dependencies in Multiple Languages. *In Proceedings of the Thirteenth Conference on Computational Natural Language Learning : Shared Task*, CoNLL '09, pages 1–18, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-29-9. URL http://dl.acm.org/citation.cfm?id=1596409.1596411. *[Cité page 83]*

Xiaodong He : Using Word-Dependent Transition Models in HMM-Based Word Alignment for Statistical Machine Translation. *In Proceedings of the Second Workshop on Statistical Machine Translation*, pages 80–87, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W07/W07-0211. *[Cité page 136]*

Teresa Herrmann, Jan Niehues et Alex Waibel : Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation. *In Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 39–47, Atlanta, Georgia, June 2013a. Association for Computational Linguistics. *[Cité pages 151 et 152]*

Teresa Herrmann, Jochen Weiner, Jan Niehues et Alex Waibel : Analyzing the Potential of Source Sentence Reordering in Statistical Machine Translation. *In Proceedings of International Workshop on Spoken Language Translation*, IWSLT, Heidelberg, Germany, Dezember 2013b. *[Cité pages 158, 176, 182, 183, et 191]*

Mark Hopkins et Jonathan May : Tuning As Ranking. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1352–1362, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL http://dl.acm.org/citation.cfm?id=2145432.2145575. *[Cité page 203]*

Susan Howlett et Mark Dras : Dual-Path Phrase-Based Statistical Machine Translation. *In Proceedings of the Australasian Language Technology Association Workshop*, pages 32–40, 2010. *[Cité page 151]*

Susan Howlett et Mark Dras : Clause Restructuring For SMT Not Absolutely Helpful. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, ACL-HLT, pages 384–388. Association for Computational Linguistics, 2011. *[Cité page 151]*

Fei Huang et Cezar Pendus : Generalized Reordering Rules for Improved SMT. *In Proceedings of the Annual Meeting on Association for Computational Linguistics*, volume 25380 de *ACL*, pages 387–392, Sofia, Bulgaria, 2013. The Association for Computer Linguistics. *[Cité page 156]*

Liang Huang : Advanced Dynamic Programming in Semiring and Hypergraph Frameworks. *In Coling 2008 : Advanced Dynamic Programming in Computational Linguistics : Theory, Algorithms and Applications - Tutorial notes*, pages 1–18, Manchester, UK, August 2008. Coling 2008 Organizing Committee. URL http://www.aclweb.org/anthology/C08-5001. *[Cité page 29]*

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas et Okan Kolak : Bootstrapping Parsers via Syntactic Projection Across Parallel Texts. *Natural Language Engineering*, 11(3):311–325, septembre 2005. *[Cité page 63]*

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh et Hajime Tsukada : Automatic Evaluation of Translation Quality for Distant Language Pairs. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 944–952. Association for Computational Linguistics, 2010a. *[Cité page 177]*

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada et Kevin Duh : Head Finalization : A Simple Reordering Rule for SOV Languages. *In Proceedings of the Joint Workshop on Statistical Machine Translation and Metrics*, WMT, pages 244–251. Association for Computational Linguistics, 2010b. *[Cité page 150]*

Rong Jin et Zoubin Ghahramani : Learning with Multiple Labels. *In* S. Thrun et K. Obermayer, éditeurs : *Processings of Advances in Neural Information Processing Systems 15*, NIPS'02, pages 897–904. MIT Press, Cambridge, MA, 2002. *[Cité pages 75, 76, et 77]*

Daniel Jurafsky et James H. Martin : *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009. ISBN 0131873210. *[Cité pages 28 et 113]*

Jun'ichi Kazama et Kentaro Torisawa : A New Perceptron Algorithm for Sequence Labeling with Non-Local Features. *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL'07, pages 315–324, 2007. *[Cité page 80]*

Maurice G. Kendall : *Rank Correlation Methods*. Theory and applications of rank order-statistics. Hafner Pub. Co., 1962. *[Cité page 159]*

Maxim Khalilov, José AR Fonollosa et Mark Dras : A new subtree-transfer approach to syntax-based reordering for statistical machine translation. *In Proceedings of the Annual Conference of the European Association for Machine Translation*, EAMT, pages 198–204, 2009. *[Cité page 150]*

Maxim Khalilov et Khalil Sima'an : Context-Sensitive Syntactic Source-Reordering by Statistical Transduction. *In Proceedings of the International Joint Conference on Natural Language Processing*, CoNLL, pages 38–46. Asian Federation of Natural Language Processing, 2011. *[Cité page 150]*

Maxim Khalilov et Khalil Sima'an : Statistical Translation After Source Reordering : Oracles, Context-Aware Models, and Empirical Analysis. *Natural Language Engineering*, 18:491–519, 10 2012. *[Cité pages 150, 153, et 178]*

Sungchul Kim, Kristina Toutanova et Hwanjo Yu : Multilingual Named Entity Recognition Using Parallel Data and Metadata from Wikipedia. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Long Papers*, ACL '12, pages 694–702, Stroudsburg, PA, USA, 2012. *[Cité page 63]*

Yoon Kim, Yacine Jernite, David Sontag et Alexander M. Rush : Character-Aware Neural Language Models. *ArXiv e-prints*, août 2015. *[Cité page 205]*

Dan Klein et Christopher D. Manning : Conditional Structure Versus Conditional Estimation in NLP Models. *In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP, pages 9–16, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1118693.1118695. *[Cité pages 39, 42, et 55]*

Kevin KNIGHT : Decoding Complexity in Word-replacement Translation Models. *Computational Linguistics*, 25(4):607–615, décembre 1999. *[Cité page 148]*

Philipp KOEHN : *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st édition, 2010. ISBN 0521874157, 9780521874151. *[Cité pages 127, 136, 138, et 139]*

Philipp KOEHN : What Is a Better Translation ? Reflections on Six Years of Running Evaluation Campaigns. *In Tralogy*, 2011. *[Cité page 138]*

Philipp KOEHN, Hieu HOANG, Alexandra BIRCH, Chris CALLISON-BURCH, Marcello FEDERICO, Nicola BERTOLDI, Brooke COWAN, Wade SHEN, Christine MORAN, Richard ZENS, Chris DYER, Ondřej BOJAR, Alexandra CONSTANTIN et Evan HERBST : Moses : Open Source Toolkit for Statistical Machine Translation. *In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. *[Cité pages 85, 135, et 139]*

Terry KOO, Xavier CARRERAS et Michael COLLINS : Simple Semi-supervised Dependency Parsing. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, ACL'08, pages 595–603, Columbus, Ohio, June 2008. Association for Computational Linguistics. *[Cité page 84]*

Mikhail KOZHEVNIKOV et Ivan TITOV : Cross-lingual Transfer of Semantic Role Labeling Models. *In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics : Long Papers*, ACL'13, pages 1190–1200, Sofia, Bulgaria, August 2013. *[Cité page 63]*

Mikhail KOZHEVNIKOV et Ivan TITOV : Cross-lingual Model Transfer Using Feature Representation Projection. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, pages 579–585, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P14/P14-2095. *[Cité page 201]*

Shankar KUMAR et William BYRNE : Local Phrase Reordering Models for Statistical Machine Translation. *In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 161–168, Vancouver, British Columbia, Canada, 2005. Association for Computational Linguistics. *[Cité pages 152 et 158]*

Ophélie LACROIX, Lauriane AUFRANT, Guillaume WISNIEWSKI et François YVON : Frustratingly Easy Cross-Lingual Transfer for Transition-Based Dependency Parsing. *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 1058–1063, San Diego, California, June 2016a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N16-1121. *[Cité page 200]*

Ophélie Lacroix, Guillaume Wisniewski et François Yvon : Cross-lingual Dependency Transfer : What Matters ? Assessing the Impact of Pre- and Post-processing. *In Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP*, pages 20–29, San Diego, California, June 2016b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W16-1203. *[Cité page 197]*

John Lafferty, Andrew McCallum et Fernando Pereira : Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data. *In Proceedings of the 18th International Conference on Machine Learning*, ICML'01, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001. *[Cité pages 28 et 78]*

Adrien Lardilleux et Yves Lepage : Sampling-based Multilingual Alignment. *In Proceedings of the International Conference RANLP-2009*, pages 214–218, Borovets, Bulgaria, September 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/R09-1040. *[Cité page 136]*

Adrien Lardilleux, François Yvon et Yves Lepage : Generalizing Sampling-based Multilingual Alignment. *Machine Translation*, 27(1):1–23, mars 2013. URL http://dx.doi.org/10.1007/s10590-012-9126-0. *[Cité page 136]*

Julia A. Lasserre, Christopher M. Bishop et Thomas P. Minka : Principled Hybrids of Generative and Discriminative Models. *In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR, pages 87–94, Washington, DC, USA, 2006. IEEE Computer Society. URL http://dx.doi.org/10.1109/CVPR.2006.227. *[Cité pages 36, 38, et 42]*

Thomas Lavergne, Alexandre Allauzen et François Yvon : Un cadre d'apprentissage intégralement discriminant pour la traduction statistique. *In Proceedings of TALN 2013 (Volume 1 : Long Papers)*, pages 450–463, Les Sables d'Olonne, France, June 2013a. ATALA. URL http://www.aclweb.org/anthology/F13-1033. *[Cité page 28]*

Thomas Lavergne, Alexandre Allauzen et François Yvon : Un cadre d'apprentissage intégralement discriminant pour la traduction statistique. *In Proceedings of TALN 2013 (Volume 1 : Long Papers)*, pages 450–463. ATALA, 2013b. URL http://aclweb.org/anthology/F13-1033. *[Cité pages 203 et 204]*

Thomas Lavergne, Olivier Cappé et François Yvon : Practical Very Large Scale CRFs. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513. Association for Computational Linguistics, 2010a. URL http://aclweb.org/anthology/P10-1052. *[Cité page 51]*

Thomas Lavergne, Olivier Cappé et François Yvon : Practical Very Large Scale CRFs. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL'11, pages 504–513, Uppsala, Sweden, 2010b. *[Cité pages 85 et 160]*

Thomas Lavergne, Josep Maria Crego, Alexandre Allauzen et François Yvon : From N-gram-based to CRF-based Translation Models. *In Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 542–553, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-12-1. URL http://dl.acm.org/citation.cfm?id=2132960.2133035. *[Cité page 203]*

Uri Lerner et Slav Petrov : Source-Side Classifier Preordering for Machine Translation. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 513–523, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. *[Cité page 150]*

Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou et Yi Guan : A Probabilistic Approach to Syntax-based Reordering for Statistical Machine Translation. *In Proceedings of the Annual Meeting on Association for Computational Linguistics*, ACL, pages 720–727. Association for Computational Linguistics, 2007. *[Cité page 150]*

Shen Li, João Graça et Ben Taskar : Wiki-ly Supervised Part-of-Speech Tagging. *In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1389–1398. Association for Computational Linguistics, 2012a. URL http://aclweb.org/anthology/D12-1127. *[Cité pages 20 et 95]*

Shen Li, João V. Graça et Ben Taskar : Wiki-ly Supervised Part-of-speech Tagging. *In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1389–1398, Stroudsburg, PA, USA, 2012b. *[Cité pages 62, 69, 70, 85, 88, 96, et 121]*

Wen Li, Lixin Duan, Ivor Wai-Hung Tsang et Dong Xu : Co-labeling : A New Multi-view Learning Approach for Ambiguous Problems. *In* Mohammed Javeed Zaki, Arno Siebes, Jeffrey Xu Yu, Bart Goethals, Geoffrey I. Webb et Xindong Wu, éditeurs : *Processings of the IEEE 12th International Conference on Data Mining*, ICDM, pages 419–428. IEEE Computer Society, 2012c. *[Cité pages 75 et 76]*

Xin Li, Paul Morie et Dan Roth : Identification and Tracing of Ambiguous Names : Discriminative and Generative Approaches. *In Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI, pages 419–424. AAAI Press, 2004. URL http://dl.acm.org/citation.cfm?id=1597148.1597217. *[Cité page 28]*

Percy Liang, Alexandre Bouchard-Côté, Dan Klein et Ben Taskar : An End-to-end Discriminative Approach to Machine Translation. *In Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 761–768, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1220175.1220271. *[Cité pages 31 et 202]*

Percy Liang et Michael I. Jordan : An Asymptotic Analysis of Generative, Discriminative, and Pseudolikelihood Estimators. *In Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 584–591, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. URL http://doi.acm.org/10.1145/1390156.1390230. *[Cité page 36]*

Wang Ling, Isabel Trancoso, Chris Dyer et Alan W Black : Character-based Neural Machine Translation. *ArXiv e-prints*, novembre 2015. *[Cité page 205]*

Lemao Liu et Liang Huang : Search-Aware Tuning for Machine Translation. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1942–1952. Association for Computational Linguistics, 2014. URL http://aclweb.org/anthology/D14-1209. *[Cité page 203]*

Thang Luong, Hieu Pham et Christopher D. Manning : Effective Approaches to Attention-based Neural Machine Translation. *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL http://aclweb.org/anthology/D15-1166. *[Cité page 205]*

Mohamed Maamouri et Ann Bies : Developing an Arabic Treebank : Methods, Guidelines, Procedures, and Tools. *In Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Semitic '04, pages 2–9, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1621804.1621808. *[Cité page 83]*

Christopher D. Manning : Part-of-Speech Tagging from 97% to 100% : Is It Time for Some Linguistics ? *In Proceedings of the Computational Linguistics and Intelligent Text Processing, 12th International Conference*, CICLing 2011, pages 171–189. Springer, 2011. *[Cité page 61]*

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz et Britta Schasberger : The Penn Treebank : Annotating Predicate Argument Structure. *In Proceedings of the Workshop on Human Language Technology*, HLT, pages 114–119, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1075812.1075835. *[Cité page 20]*

José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrick Lambert, José A.R. Fonollosa et Marta R. Costa-Jussà : N-gram-based Machine Translation. *Computational Linguistics*, 32(4):527–549, 2006. *[Cité page 141]*

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló et Jungmee Lee : Universal Dependency Annotation for Multilingual Parsing. *In Proceedings of the 51st*

Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers), pages 92–97, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P13-2017. *[Cité page 83]*

Tom Minka : Discriminative models, not discriminative training. Rapport technique, October 2005. URL https://www.microsoft.com/en-us/research/publication/discriminative-models-not-discriminative-training/. *[Cité pages 36, 38, et 41]*

Robert Moore : Fast High-Accuracy Part-of-Speech Tagging by Independent Classifiers. *In Proceedings of the 25th International Conference on Computational Linguistics : Technical Papers*, COLING'14, pages 1165–1176, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. *[Cité pages 79 et 121]*

Thomas Mueller, Helmut Schmid et Hinrich Schütze : Efficient Higher-Order CRFs for Morphological Tagging. *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1032. *[Cité page 104]*

Thomas Müller, Helmut Schmid et Hinrich Schütze : Efficient Higher-Order CRFs for Morphological Tagging. *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, numéro October, pages 322–332, Seattle, Washington, USA, 2013. Association for Computational Linguistics. *[Cité pages 113, 116, 117, 118, 122, et 199]*

Bernard Mérialdo : Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2):155–171, juin 1994. *[Cité pages 61, 62, et 121]*

Tahira Naseem, Benjamin Snyder, Jacob Eisenstein et Regina Barzilay : Multilingual Part-of-Speech Tagging : Two Unsupervised Approaches. *Journal of Artificial Intelligence Research*, 36, November 2009. *[Cité page 121]*

Rebecca Nesson, Stuart M. Shieber et Alexander Rush : Induction of Probabilistic Synchronous Tree-Insertion Grammars for Machine Translation. *In 5th Conference of the Association for Machine Translation in the Americas (AMTA)*, Boston, Massachusetts, August 2006. URL http://www.mt-archive.info/AMTA-2006-Nesson.pdf. *[Cité page 133]*

Graham Neubig, Taro Watanabe et Shinsuke Mori : Inducing a Discriminative Parser to Optimize Machine Translation Reordering. *In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL, pages 843–853. Association for Computational Linguistics, 2012. *[Cité pages 151, 153, 159, et 177]*

Andrew Y. Ng et Michael I. Jordan : On Discriminative vs. Generative Classifiers : A comparison of logistic regression and naive Bayes. *In Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press, 2002. *[Cité page 36]*

Jan Niehues, Teresa Herrmann, Stephan Vogel et Alex Waibel : Wider Context by Using Bilingual Language Models in Machine Translation. *In Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 198–206, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-2124. *[Cité page 144]*

Jan Niehues et Muntsin Kolss : A POS-Based Model for Long-Range Reorderings in SMT. *In Proceedings of the Workshop on Statistical Machine Translation*, StatMT, pages 206–214. Association for Computational Linguistics, 2009. *[Cité pages 151 et 152]*

Joakim Nivre, Željko Agić, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Cristina Bosco, Sam Bowman, Giuseppe G. A. Celano, Miriam Connor, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Daniel Galbraith, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Berta Gonzales, Bruno Guillaume, Jan Hajič, Dag Haug, Radu Ion, Elena Irimia, Anders Johannsen, Hiroshi Kanayama, Jenna Kanerva, Simon Krek, Veronika Laippala, Alessandro Lenci, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Shunsuke Mori, Hanna Nurmi, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Prokopis Prokopidis, Sampo Pyysalo, Loganathan Ramasamy, Rudolf Rosa, Shadi Saleh, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Kiril Simov, Aaron Smith, Jan Štěpánek, Alane Suhr, Zsolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Sumire Uematsu, Larraitz Uria, Viktor Varga, Veronika Vincze, Zdeněk Žabokrtský, Daniel Zeman et Hanzhi Zhu : Universal Dependencies 1.2, 2015. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague. *[Cité page 65]*

Franz Josef Och : Minimum Error Rate Training in Statistical Machine Translation. *In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1075096.1075117. *[Cité page 140]*

Franz Josef Och et Hermann Ney : Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. *In Proceedings of the 40th Annual Meeting*

on Association for Computational Linguistics, ACL '02, pages 295–302, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1073083.1073133. *[Cité pages 131, 132, et 202]*

Franz Josef Och et Hermann Ney : A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 2003. *[Cité pages 67 et 136]*

Robert Östling : Stagger : an Open-Source Part of Speech Tagger for Swedish. *Northern European Journal of Language Technology (NEJLT)*, 3:1–18, 2013. *[Cité page 121]*

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider et Noah A. Smith : Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. *In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, NAACL'13, pages 380–390, Atlanta, Georgia, June 2013. *[Cité page 84]*

Sebastian Padó et Mirella Lapata : Cross-lingual Annotation Projection of Semantic Roles. *Journal of Artificial Intelligence Research*, 36(1):307–340, septembre 2009. *[Cité page 63]*

Kishore Papineni, Salim Roukos, Todd Ward et Wei-Jing Zhu : BLEU : A Method for Automatic Evaluation of Machine Translation. *In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1073083.1073135. *[Cité page 138]*

Slav Petrov, Dipanjan Das et Ryan McDonald : A Universal Part-of-Speech Tagset. *In* Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk et Stelios Piperidis, éditeurs : *Proceedings of the Eight International Conference on Language Resources and Evaluation*, LREC'12, Istanbul, Turkey, may 2012a. European Language Resources Association (ELRA). *[Cité pages 65 et 84]*

Slav Petrov, Dipanjan Das et Ryan McDonald : A Universal Part-of-Speech Tagset. *In* Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk et Stelios Piperidis, éditeurs : *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2089–2096, Istanbul, Turkey, May 2012b. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7. URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/274_Paper.pdf. ACL Anthology Identifier : L12-1115. *[Cité pages 156 et 161]*

Prokopis Prokopidis, Elina Desypri, Maria Koutsombogera, Haris Papageorgiou et Stelios Piperidis : Theoretical and Practical Issues in the Construction of a Greek Dependency Treebank. *In* Montserrat Civit, Sandra Kubler et Ma. Antonia Marti, éditeurs : *Proceedings of The Fourth Workshop on Treebanks and Linguistic Theories*,

TLT'05, pages 149–160, Barcelona, Spain, December 2005. Universitat de Barcelona. *[Cité page 83]*

Ananthakrishnan Ramanathan et Karthik Visweswariah : A Study of Word-Classing for MT Reordering. *In Proceedings of the International Conference on Language Resources and Evaluation*, LREC, pages 3971–3976, Istanbul, Turkey, 2012. *[Cité page 156]*

Lev Ratinov et Dan Roth : Design Challenges and Misconceptions in Named Entity Recognition. *In Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. *[Cité page 80]*

Adwait Ratnaparkhi : A Maximum Entropy Model for Part-Of-Speech Tagging. *In Proceedings of the 1996 Conference on Empirical Methods in Natural Language Processing*, EMNLP'96. Association for Computational Linguistics, 1996. *[Cité pages 79 et 121]*

Adwait Ratnaparkhi : A simple introduction to maximum entropy models for natural language processing. Rapport technique, nstitute for Research in Cognitive Science, University of Pennsylvania, 1997. *[Cité page 28]*

Sujith Ravi et Kevin Knight : Minimized Models for Unsupervised Part-of-speech Tagging. *In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL '09, pages 504–512, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-45-9. URL http://dl.acm.org/citation.cfm?id=1687878.1687950. *[Cité page 121]*

Philip Resnik et Noah A. Smith : The Web As a Parallel Corpus. *Computational Linguistics*, 29(3):349–380, septembre 2003. *[Cité page 63]*

Martin Riedmiller et Heinrich Braun : A Direct Adaptive Method for Faster Backpropagation Learning : The RPROP Algorithm. *In Proceedings of the IEEE International Conference on Neural Networks*, ICNN '93, pages 586–591, 1993. *[Cité pages 51 et 109]*

Brian Roark, Murat Saraclar, Michael Collins et Mark Johnson : Discriminative Language Modeling with Conditional Random Fields and the Perceptron Algorithm. *In Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, ACL, pages 47–54, Barcelona, Spain, July 2004. URL http://www.aclweb.org/anthology/P04-1007. *[Cité page 42]*

Stéphane Ross et Drew Bagnell : Efficient Reductions for Imitation Learning. *In Proceedings of the International Conference on Artificial Intelligence on Statistics*, AISTATS'10, pages 661–668, 2010. *[Cité page 80]*

Kay Rottmann et Stephan Vogel : Word reordering in statistical machine translation with a POS-based distortion model. *Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 171–180, 2007. *[Cité page 150]*

Helmut Schmid : Probabilistic Part-of-Speech Tagging Using Decision Trees. *In Proceedings of International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK, 1994. *[Cité page 160]*

Miikka Silfverberg, Teemu Ruokolainen, Krister Lindén et Mikko Kurimo : Part-of-Speech Tagging using Conditional Random Fields : Exploiting Sub-Label Dependencies for Improved Accuracy. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 259–264. Association for Computational Linguistics, 2014. *[Cité page 117]*

A. Noah Smith et Jason Eisner : Contrastive Estimation : Training Log-Linear Models on Unlabeled Data. *In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 354–362. Association for Computational Linguistics, 2005. *[Cité pages 48, 122, et 199]*

Noah A. Smith : *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, May 2011. *[Cité pages 36, 48, et 50]*

Noah A. Smith : Adversarial Evaluation for Models of Natural Language. *CoRR*, abs/1207.0245, 2012. URL http://arxiv.org/abs/1207.0245. *[Cité page 138]*

Noah A Smith, David A Smith et Roy W Tromble : Context-Based Morphological Disambiguation with Random Fields. *In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 475–482, Vancouver, British Columbia, Canada, 2005. Association for Computational Linguistics. *[Cité pages 121 et 122]*

Matthew Snover, Bonnie J. Dorr, Richard Schwartz, Linnea Micciulla et John Makhoul : A Study of Translation Edit Rate with Targeted Human Annotation. *In 5th Conference of the Association for Machine Translation in the Americas*, Boston, Massachusetts, August 2006. URL http://mt-archive.info/AMTA-2006-Snover.pdf. *[Cité page 138]*

Artem Sokolov, Guillaume Wisniewski et François Yvon : Computing Lattice BLEU Oracle Scores for Machine Translation. *In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 120–129, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. ISBN 978-1-937284-19-0. URL http://dl.acm.org/citation.cfm?id=2380816.2380834. *[Cité pages 144 et 178]*

Artem Sokolov, Guillaume Wisniewski et Franccois Yvon : Lattice BLEU Oracles in Machine Translation. *ACM Trans. Speech Lang. Process.*, 10(4):18 :1–18 :29, janvier 2014. ISSN 1550-4875. URL http://doi.acm.org/10.1145/2513147. *[Cité page 144]*

Miloš Stanojević et Khalil Sima'an : BEER : BEtter Evaluation as Ranking. *In Proceedings of the Workshop on Statistical Machine Translation*, WMT, pages 414–419, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. *[Cité page 192]*

Miloš Stanojević et Khalil Sima'an : Reordering Grammar Induction. *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 44–54, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL http://aclweb.org/anthology/D15-1005. *[Cité page 201]*

Milan Straka et Jana Straková : Czech Models (MorfFlex CZ + PDT) for MorphoDiTa, 2013. *[Cité page 160]*

Jana Straková, Milan Straka et Jan Hajič : Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. *In Proceedings of Annual Meeting of the Association for Computational Linguistics : System Demonstrations*, pages 13–18, Baltimore, Maryland, June 2014. Association for Computational Linguistics. *[Cité page 160]*

Ilya Sutskever, Oriol Vinyals et Quoc V Le : Sequence to Sequence Learning with Neural Networks. *In Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf. *[Cité page 204]*

Charles Sutton et Andrew McCallum : An Introduction to Conditional Random Fields for Relational Learning. *In* Lise Getoor et Ben Taskar, éditeurs : *Introduction to Statistical Relational Learning*. MIT Press, 2007. *[Cité page 78]*

Charles Sutton et Andrew McCallum : An Introduction to Conditional Random Fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012. *[Cité pages 28, 41, et 46]*

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald et Joakim Nivre : Token and Type Constraints for Cross-Lingual Part-of-Speech Tagging. *Transactions of the Association of Computational Linguistics – Volume 1*, pages 1–12, 2013a. URL http://aclweb.org/anthology/Q13-1001. *[Cité pages 20, 59, 95, 105, 118, et 119]*

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald et Joakim Nivre : Token and Type Constraints for Cross-Lingual Part-of-Speech Tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12, 2013b. *[Cité pages 59, 63, 64, 65, 67, 72, 73, 74, 75, 77, 79, 84, 85, 86, 96, et 121]*

Oscar Täckström, Ryan McDonald et Jakob Uszkoreit : Cross-lingual Word Clusters for Direct Transfer of Linguistic Structure. *In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, NAACL HLT'12, pages 477–487, 2012. *[Cité page 84]*

David Talbot, Hideto Kazawa, Hiroshi Ichikawa, Jason Katz-Brown, Masakazu Seno et Franz J. Och : A Lightweight Evaluation Framework for Machine Translation Reordering. *In Proceedings of the Workshop on Statistical Machine Translation*, WMT, pages 12–21. Association for Computational Linguistics, 2011. *[Cité pages 159 et 177]*

Isabelle Tellier et Marc Tommasi : Champs Markoviens Conditionnels pour l'extraction d'information. *In* Éric Gaussier et François Yvon, éditeurs : *Modèles probabilistes pour l'accès à l'information textuelle*, pages 223–267. Hermès, 2011. *[Cité pages 28 et 78]*

Jörg Tiedemann : Rediscovering Annotation Projection for Cross-Lingual Parser Induction. *In Proceedings of the 25th International Conference on Computational Linguistics : Technical Papers*, COLING'14, pages 1854–1864, Dublin, Ireland, 2014. Dublin City University and Association for Computational Linguistics. *[Cité pages 64 et 101]*

Christoph Tillmann : A Unigram Orientation Model for Statistical Machine Translation. *In Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, pages 101–104, 2004. *[Cité pages 141 et 149]*

Christoph Tillmann et Tong Zhang : A Discriminative Global Training Algorithm for Statistical MT. *In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 721–728, Sydney, Australia, July 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P06-1091. *[Cité pages 31 et 203]*

Nadi Tomeh : *Discriminative Alignment Models For Statistical Machine Translation.* Thèse de doctorat, University of Paris-Sud, Orsay, France, 2012. URL https://tel.archives-ouvertes.fr/tel-00720250. *[Cité page 134]*

Kristina Toutanova, H. Tolga Ilhan et Christopher D. Manning : Extentions to HMM-based Statistical Word Alignment Models. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 87–94, Philadelphia, July 2002. Association for Computational Linguistics. URL http://acl.ldc.upenn.edu/W/W02/W02-1012.pdf. *[Cité page 136]*

Kristina Toutanova et Mark Johnson : A Bayesian LDA-based model for semi-supervised part-of-speech tagging. *In Proceedings of the Neural Information Processing Systems*, NIPS'07, pages 1521–1528, 2007. *[Cité page 62]*

Roy Tromble et Jason Eisner : Learning Linear Ordering Problems for Better Translation. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 1007–1016, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. *[Cité pages 149, 151, 153, et 201]*

Roy Tromble, Shankar Kumar, Franz Och et Wolfgang Macherey : Lattice Minimum Bayes-Risk Decoding for Statistical Machine Translation. *In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 620–629, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D08-1065. *[Cité pages 144 et 145]*

Yuta Tsuboi, Hisashi Kashima, Hiroki Oda, Shinsuke Mori et Yuji Matsumoto : Training Conditional Random Fields Using Incomplete Annotations. *In Proceedings of the 22nd International Conference on Computational Linguistics*, volume 1 de *COLING'08*, pages 897–904, 2008. *[Cité pages 75, 76, et 77]*

Yoshimasa Tsuruoka, Yusuke Miyao et Jun'ichi Kazama : Learning with Lookahead : Can History-Based Models Rival Globally Optimized Models ? *In Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL'11, pages 238–246, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. *[Cité pages 79 et 80]*

Akira Ushioda : Hierarchical Clustering of Words. *In Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, COLING, pages 1159–1162, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/993268.993390. *[Cité page 201]*

Lonneke van der Plas, Marianna Apidianaki et Chenhua Chen : Global Methods for Cross-lingual Semantic Role and Predicate Labelling. *In Proceedings of the 25th International Conference on Computational Linguistics : Technical Papers*, COLING'14, pages 1279–1290, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. *[Cité page 63]*

Karthik Visweswariah, Mitesh M. Khapra et Ananthakrishnan Ramanathan : Cut the noise : Mutually reinforcing reordering and alignments for improved machine translation. *In Proceedings of the Annual Meeting on Association for Computational Linguistics*, ACL, pages 1275–1284, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. *[Cité page 151]*

Karthik Visweswariah, Rajakrishnan Rajkumar, Ankur Gandhe, Ananthakrishnan Ramanathan et Jiri Navratil : A Word Reordering Model for Improved Machine

Translation. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 486–496, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. *[Cité pages 151, 174, et 201]*

Stephan Vogel, Hermann Ney et Christoph Tillmann : HMM-based Word Alignment in Statistical Translation. *In Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, COLING, pages 836–841, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/993268.993313. *[Cité page 136]*

Martin J. Wainwright et Michael I. Jordan : Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, janvier 2008. URL http://dx.doi.org/10.1561/2200000001. *[Cité pages 28, 36, 37, 45, et 47]*

Mengqiu Wang et Christopher D. Manning : Cross-lingual Projected Expectation Regularization for Weakly Supervised Learning. *Transactions of the Association for Computational Linguistics*, 2:55–66, février 2014a. *[Cité page 63]*

Mengqiu Wang et Christopher D Manning : Cross-lingual Projected Expectation Regularization for Weakly Supervised Learning. *Transaction of the ACL*, 2(1):55–66, 2014b. *[Cité pages 64 et 200]*

Jakub Waszczuk : Harnessing the CRF Complexity with Domain-Specific Constraints. The Case of Morphosyntactic Tagging of a Highly Inflected Language. *In Proceedings of COLING 2012*, numéro December 2012, pages 2789–2804, Mumbai, India, 2012. The COLING 2012 Organizing Committee. *[Cité page 121]*

Guillaume Wisniewski, Alexandre Allauzen et François Yvon : Assessing Phrase-Based Translation Models with Oracle Decoding. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 933–943. Association for Computational Linguistics, 2010. *[Cité page 178]*

Guillaume Wisniewski et François Yvon : Oracle Decoding as a New Way to Analyze Phrase-based Machine Translation. *Machine Translation*, 28(2):1–24, 2013. *[Cité pages 152 et 178]*

David H. Wolpert : Stacked Generalization. *Neural Networks*, 5, 1992. *[Cité page 80]*

Dekai Wu : Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377–403, 1997. *[Cité pages 137 et 152]*

Fei Xia et Michael McCord : Improving a Statistical MT System with Automatically Learned Rewrite Patterns. *In Proceedings of the International Conference on Computational Linguistics*, COLING, pages 508–514, Geneva, Switzerland, 2004. *[Cité pages 149 et 150]*

Peng Xu, Jaeho Kang, Michael Ringgaard et Franz Och : Using a Dependency Parser to Improve SMT for Subject-object-verb Languages. *In Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, pages 245–253. Association for Computational Linguistics, 2009. *[Cité page 150]*

Kenji Yamada et Kevin Knight : A Syntax-based Statistical Translation Model. *In Proceedings of the Annual Meeting on Association for Computational Linguistics*, ACL, pages 523–530. Association for Computational Linguistics, 2001. *[Cité page 152]*

David Yarowsky, Grace Ngai et Richard Wicentowski : Inducing Multilingual Text Analysis Tools via Robust Projection Across Aligned Corpora. *In Proceedings of the First International Conference on Human Language Technology Research*, HLT '01, pages 1–8, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics. *[Cité pages 20, 63, et 65]*

Victor H. Yngve : The technical feasibility of translating languages by machine. *In Communication and Electronics 28*, pages 792–797, 1957. *[Cité page 16]*

Daniel Zeman et Philip Resnik : Cross-Language Parser Adaptation between Related Languages. *In Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42, Hyderabad, India, January 2008. *[Cité page 63]*

Othman Zennaki, Nasredine Semmar et Laurent Besacier : Utilisation des réseaux de neurones récurrents pour la projection interlingue d'étiquettes morpho-syntaxiques à partir d'un corpus parallèle. *In Actes de la 22e conférence sur le Traitement Automatique des Langues Naturelles*, pages 529–536, Caen, France, June 2015. Association pour le Traitement Automatique des Langues. URL http://www.atala.org/taln_archives/TALN/TALN-2015/taln-2015-court-032. *[Cité page 200]*

Richard Zens et Hermann Ney : A Comparative Study on Reordering Constraints in Statistical Machine Translation. *In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 144–151, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1075096.1075115. *[Cité pages 137 et 152]*

Richard Zens, Hermann Ney, Taro Watanabe et Eiichiro Sumita : Reordering Constraints for Phrase-Based Statistical Machine Translation. *In Proceedings of the International Conference on Computational Linguistics*, COLING, pages 205–211, Geneva, Switzerland, 2004. *[Cité pages 137 et 152]*

Richard Zens, Franz Joseph Och et Herman Ney : Phrase-based statistical machine translation. *In* M. Jarke, J. Koehler et G. Lakemeyer, éditeurs : *Lecture Notes in Artificial Intelligence*, volume 2479 de *LNAI*, pages 18–32. Springer Verlag, 2002. *[Cité pages 141 et 152]*

Dakun Zhang, Le Sun et Wenbo Li : A Structured Prediction Approach for Statistical Machine Translation. *In Proceedings of the Third International Joint Conference on Natural Language Processing : Volume-II*, 2008. URL http://aclweb.org/anthology/I08-2087. *[Cité page 31]*

Xiagn Zhang, Junbo Zhao et Yann LeCun : Character-level Convolutional Networks for Text Classification. *ArXiv e-prints*, septembre 2015. *[Cité page 205]*

Yuan Zhang, David Gaddy, Regina Barzilay et Tommi Jaakkola : Ten Pairs to Tag – Multilingual POS Tagging via Coarse Mapping between Embeddings. *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 1307–1317, San Diego, California, June 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N16-1156. *[Cité page 201]*

Yuan Zhang, Roi Reichart, Regina Barzilay et Amir Globerson : Learning to Map into a Universal POS Tagset. *In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1368–1378, Stroudsburg, PA, USA, 2012. *[Cité page 98]*

Yuqi Zhang, Richard Zens et Hermann Ney : Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation. *In Proceedings of the Workshop on Syntax and Structure in Statistical Translation*, SSST-NAACL, pages 1–8, Rochester, New York, April 2007. Association for Computational Linguistics. *[Cité page 150]*

Hai Zhao, Yan Song, Chunyu Kit et Guodong Zhou : Cross-Language Dependency Parsing Using a Bilingual Lexicon. *In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL '09, pages 55–63, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-45-9. *[Cité page 63]*

Simon Zwarts et Mark Dras : This Phrase-Based SMT System is Out of Order : Generalised Word Reordering in Machine Translation. *In Proceedings of the Australasian Language Technology Workshop*, pages 149–156, 2006. *[Cité page 151]*

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

Titre : Modèles exponentiels et contraintes sur les espaces de recherche en traduction automatique et pour le transfert cross-lingue

Mots clés : Traduction automatique; Contraintes de réordonnancements; Étiquetage morpho-syntaxique; Transfert cross-lingue; Apprentissage faiblement supervisé; Champs markoviens aléatoires

Résumé : La plupart des méthodes de traitement automatique des langues (TAL) peuvent être formalisées comme des problèmes de prédiction, dans lesquels on cherche à choisir automatiquement l'hypothèse la plus plausible parmi un très grand nombre de candidats. Malgré de nombreux travaux qui ont permis de mieux prendre en compte la structure de l'ensemble des hypothèses, la taille de l'espace de recherche est généralement trop grande pour permettre son exploration exhaustive. Dans ce travail, nous nous intéressons à l'importance du design de l'espace de recherche et étudions l'utilisation de contraintes pour en réduire la taille et la complexité. Nous nous appuyons sur l'étude de trois problèmes linguistiques — l'analyse morpho-syntaxique, le transfert cross-lingue et le problème du réordonnancement en traduction — pour mettre en lumière les risques, les avantages et les enjeux du choix de l'espace de recherche dans les problèmes de TAL. Par exemple, lorsque l'on dispose d'informations a priori sur les sorties possibles d'un problème d'apprentissage structuré, il semble naturel de les inclure dans le processus de modélisation pour réduire l'espace de recherche et ainsi permettre une accélération des traitements lors de la phase d'apprentissage. Une étude de cas sur les modèles exponentiels pour l'analyse morpho-syntaxique montre paradoxalement que cela peut conduire à d'importantes dégradations des résultats, et cela même quand les contraintes associées sont pertinentes. Parallèlement, nous considérons l'utilisation de ce type de contraintes pour généraliser le problème de l'apprentissage supervisé au cas où l'on ne dispose que d'informations partielles et incomplètes lors de l'apprentissage, qui apparaît par exemple lors du transfert cross-lingue d'annotations. Nous étudions deux méthodes d'apprentissage faiblement supervisé, que nous formalisons dans le cadre de l'apprentissage ambigu, appliquées à l'analyse morpho-syntaxiques de langues peu dotées en ressources linguistiques. Enfin, nous nous intéressons au design de l'espace de recherche en traduction automatique. Les divergences dans l'ordre des mots lors du processus de traduction posent un problème combinatoire difficile. En effet, il n'est pas possible de considérer l'ensemble factoriel de tous les réordonnancements possibles, et des contraintes sur les permutations s'avèrent nécessaires. Nous comparons différents jeux de contraintes et explorons l'importance de l'espace de réordonnancement dans les performances globales d'un système de traduction. Si un meilleur design permet d'obtenir de meilleurs résultats, nous montrons cependant que la marge d'amélioration se situe principalement dans l'évaluation des réordonnancements plutôt que dans la qualité de l'espace de recherche.

Title : Log-linear Models and Search Space Constraints in Statistical Machine Translation and Cross-lingual Transfer

Keywords : Statistical Machine Translation; Reordering Contraints; Cross-Lingual Transfer; Weakly Supervised Learning; Conditional Random Fields

Abstract : Most natural language processing tasks are modeled as prediction problems where one aims at finding the best scoring hypothesis from a very large pool of possible outputs. Even if algorithms are designed to leverage some kind of structure, the output space is often too large to be searched exaustively. This work aims at understanding the importance of the search space and the possible use of constraints to reduce it in size and complexity. We report in this thesis three case studies which highlight the risk and benefits of manipulating the seach space in learning and inference. When information about the possible outputs of a sequence labeling task is available, it may seem appropriate to include this knowledge into the system, so as to facilitate and speed-up learning and inference. A case study on type constraints for CRFs however shows that using such constraints at training time is likely to drastically reduce performance, even when these constraints are both correct and useful at decoding. On the other side, we also consider possible relaxations of the supervision space, as in the case of learning with latent variables, or when only partial supervision is available, which we cast as ambiguous learning. Such weakly supervised methods, together with cross-lingual transfer and dictionary crawling techniques, allow us to develop natural language processing tools for under-resourced languages. Word order differences between languages pose several combinatorial challenges to machine translation and the constraints on word reorderings have a great impact on the set of potential translations that is explored during search. We study reordering constraints that allow to restrict the factorial space of permutations and explore the impact of the reordering search space design on machine translation performance. However, we show that even though it might be desirable to design better reordering spaces, model and search errors seem yet to be the most important issues.

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