Apprentissage d'analyseur en dépendances cross-lingue par projection partielle de dépendances

Ophélie Lacroix[1] Lauriane Aufrant[1, 2] Guillaume Wisniewski[1] François Yvon[1]
(1) LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, F-91405 Orsay
(2) DGA, 60 boulevard du Général Martial Valin, F-75509 Paris
{lacroix, aufrant, wisniews, yvon}@limsi.fr

RÉSUMÉ

Cet article présente une méthode simple de transfert cross-lingue de dépendances. Nous montrons tout d'abord qu'il est possible d'apprendre un analyseur en dépendances par transition à partir de données partiellement annotées. Nous proposons ensuite de construire de grands ensembles de données partiellement annotés pour plusieurs langues cibles en projetant les dépendances via les liens d'alignement les plus sûrs. En apprenant des analyseurs pour les langues cibles à partir de ces données partielles, nous montrons que cette méthode simple obtient des performances qui rivalisent avec celles de méthodes état-de-l'art récentes, tout en ayant un coût algorithmique moindre.

ABSTRACT

Cross-lingual learning of dependency parsers from partially projected dependencies

This paper presents a simple strategy for transferring dependency parsers across languages. We first show that learning transition-based parser from partially annotated data is possible and effective. Then we propose to build large partially annotated dataset for several target languages via the projection of annotations through unambiguous word alignments. Based on the results obtained with such methodology, we show that our method is therefore easy to implement and compete with recent algorithmically costly methods at a much cheaper computational cost.

MOTS-CLÉS : Transfert cross-lingue, Analyse en dépendances, Annotations partielles.

KEYWORDS: Cross-lingual transfer, Dependency parsing, Partially annotated data.

1 Introduction

De multiples tâches de traitement automatique des langues reposent sur des méthodes d'apprentissage supervisé nécessitant des corpus annotés de taille suffisante. Cependant, toutes les langues ne sont pas sur un pied d'égalité lorsqu'il s'agit d'appliquer ces méthodes : certaines langues, telle que l'anglais, sont largement pourvues en ressources annotées tandis que d'autres, généralement qualifiées de « peu dotées », en manquent cruellement. Dans ce contexte, de nombreuses techniques ont été mises en œuvre dans le but de transférer des annotations (par exemple grammaticales, syntaxiques, etc.) depuis une langue source bien dotée vers une langue cible peu dotée, puis d'utiliser cette supervision imparfaite pour entraîner des systèmes de traitement de la langue cible (Pan & Yang, 2010).

Dans ce travail, nous nous intéressons plus particulièrement au transfert cross-lingue pour l'analyse en dépendances. Dans ce domaine, deux types d'approches ont été proposés : le transfert direct de

modèles et le transfert d'annotations par alignement. Le premier type s'appuie sur une représentation commune des tokens des langues sources et cibles (par exemple en utilisant les étiquettes morpho-syntaxiques universelles de Petrov *et al.* (2012)), permettant ainsi d'entraîner un analyseur sur les phrases sources et de l'appliquer directement aux phrases cibles sans tenir compte des informations lexicales (c.-à-d. des tokens). Cette technique d'« analyse délexicalisée », initialement introduite par Zeman & Resnik (2008), peut être améliorée par des méthodes d'auto-apprentissage, par la sélection intelligente des données, par la relexicalisation des données ou le transfert multi-source (Naseem *et al.*, 2010; Cohen *et al.*, 2011; Søgaard, 2011; Täckström *et al.*, 2013).

Le second type d'approche requiert l'emploi de données parallèles et repose sur la *projection* des dépendances des phrases sources aux phrases cibles au travers des liens d'alignement. Cette méthode introduite par Hwa *et al.* (2005) a été depuis reprise et améliorée (Ozdowska, 2006; Tiedemann, 2014). Une des difficultés soulevée par cette approche vient du fait que les structures syntaxiques entre les phrases sources et cibles ne sont pas isomorphes du fait des divergences syntaxiques des langues, telles l'absence/la concaténation des partitifs. En outre, les alignements peuvent être bruités et l'analyse en dépendances des phrases sources inexacte. L'apprentissage doit alors soit s'effectuer sur un faible nombre de données intégralement annotées, soit s'adapter à des données partiellement annotées. Dans ce cas, les arbres partiels sont généralement complétés, par exemple en attachant chaque mot isolé à l'aide d'une dépendance factice (Spreyer & Kuhn, 2009) ou en soumettant toutes les possibilités de rattachement de ces mots (Li *et al.*, 2014), en considérant les dépendances prédites par une analyse délexicalisée (Ma & Xia, 2014) ou par une première étape d'apprentissage sur des phrases intégralement annotées (Rasooli & Collins, 2015). Une variante récente (Tiedemann *et al.*, 2014) de cette méthode utilise des corpus parallèles synthétiques produits par traduction automatique, ce qui permet de s'affranchir du bruit d'alignement et de simplifier les règles de projection ; elle présente l'inconvénient fournir des données artificielles, de qualité difficilement contrôlable, au parseur en langue cible.

Dans cet article, nous proposons une méthode simple et efficace de transfert d'annotations qui ne nécessite pas de recourir au filtrage et aux règles de transfert de Hwa *et al.* (2005). Lors de la projection des dépendances d'une phrase source vers une phrase cible, nous transférons uniquement les dépendances entre les mots dont l'alignement est sûr, produisant de grands corpus de données partiellement annotées. Nous montrons qu'il est possible d'apprendre un analyseur par transition projectif sur ces données : ceux-ci sont en effet appris par correction d'erreur à l'aide d'un oracle dynamique (Goldberg & Nivre, 2012), ce qui permet d'ignorer les mots non attachés pour se concentrer sur la correction des dépendances connues et de minimiser l'impact de l'accumulation des erreurs.

Le reste de l'article est organisé comme suit : dans la section 2, nous détaillons le système d'analyse en dépendances qui permet l'apprentissage à partir de données partiellement annotées. Nous montrons, par des expériences sur des corpus artificiels (section 2.2), que l'apprentissage à partir de phrases partiellement annotées est possible et demeure efficace. Puis nous appliquons cette méthode dans le cadre du transfert d'annotations (section 3) en dépendances.

2 Apprentissage à partir de données partiellement annotées

Dans cette section, nous commnençons par rappeler le principe d'un analyseur *Arc-Eager* (Nivre, 2003) puis le principe de son apprentissage avec un oracle dynamique (Goldberg & Nivre, 2012). Nous expliquons ensuite en quoi ce système est adapté à l'apprentissage à partir de données incomplètes.

2.1 Analyse par transition : inférence

Un analyseur par transitions construit un arbre en dépendances de manière incrémentale ; il parcourt la phrase à analyser de gauche à droite et ajoute des dépendances au fur et à mesure par application d'actions. Dans le cas d'un analyseur *Arc-Eager* quatre actions, définies dans le tableau 1, sont possibles : LEFTARC, RIGHTARC, SHIFT et REDUCE. À chaque étape de l'analyse, le choix de l'action à appliquer pour passer à la configuration suivante est déterminé par un classifieur multi-classe. Celui-ci calcule le score de chaque action possible à ce moment de l'analyse (c.-à-d. les actions respectant les conditions du tableau 1) et choisit d'appliquer l'action a^* de plus grand score :

$$a^* = \underset{a \in \text{LEGAL}(c)}{\arg\max} \langle \phi(c,a) | \mathbf{w} \rangle, \tag{1}$$

où c est la configuration courante, $\mathbf{w}$, un vecteur de paramètres, $\phi(c,a)$, un vecteur de traits décrivant la configuration courante et $\text{LEGAL}(c)$ l'ensemble des actions qui peuvent être appliquées à partir de c. L'analyse syntaxique d'une phrase correspond alors à une suite de décisions qui permet de passer progressivement d'une configuration initiale (notée $c_0 = ([],[w_1,...,w_n],\emptyset)$), décrivant une structure de dépendance vide, à une configuration finale dans laquelle chaque mot est rattaché à sa tête.

Actions	**Effets sur les configurations**			**Conditions**
LEFTARC	$(\sigma \mid w_i,\ w_j \mid \beta,\ A)$	$\Rightarrow$	$(\sigma,\ w_j \mid \beta,\ A \cup \{(j,i)\})$	$i \neq 0 \wedge \neg\exists k\ (k,i) \in A$
RIGHTARC	$(\sigma \mid w_i,\ w_j \mid \beta,\ A)$	$\Rightarrow$	$(\sigma \mid w_j,\ \beta,\ A \cup \{(i,j)\})$	$\neg\exists k\ (k,j) \in A$
REDUCE	$(\sigma \mid w_i,\ \beta,\ A)$	$\Rightarrow$	$(\sigma,\ \beta,\ A)$	$\exists k\ (k,i) \in A$
SHIFT	$(\sigma,\ w_i \mid \beta,\ A)$	$\Rightarrow$	$(\sigma \mid w_i,\ \beta,\ A)$	

TABLE 1 – Effets et conditions d'application des actions (non étiquetées) de l'analyseur *ArcEager* sur les configurations. Une configuration est un triplet $(\sigma,\ \beta,\ A)$ où σ est une pile de mots, β est un buffer de mots non encore traités et A est un ensemble d'arcs.

2.2 Apprentissage partiel

L'algorithme 1 décrit comment est appris l'analyseur par transition : pour chaque phrase de l'ensemble d'apprentissage, un arbre de dépendances est construit de manière incrémentale comme pour l'étape d'inférence. À chaque étape, si l'action prédite crée une dépendance qui n'est pas dans l'arbre de référence ou empêche la création d'une dépendance de celui-ci, le vecteur de paramètres est mis à jour selon la règle du perceptron. Goldberg & Nivre (2012) introduisent une méthode qui construit, pour une configuration c donnée, l'ensemble, $\text{CORRECT}_y(c)$, des actions « correctes » n'empêchant pas la création d'une des dépendances de la référence y. Si plusieurs actions sont correctes, l'action choisie comme référence lors de la mise à jour est celle de plus haut score selon le modèle. En résumé, l'apprentissage consiste à construire la sortie associée à une phrase en vérifiant, à chaque étape de l'analyse, si l'action prédite (c.-à-d. l'action de plus haut score appartenant à l'ensemble $\text{LEGAL}(c)$) est différente de l'action correcte (l'action de plus haut score appartenant à l'ensemble $\text{CORRECT}_y(c)$) ; dans ce cas, le vecteur de paramètres $\mathbf{w}$ est mis à jour.

Il est important de remarquer que l'algorithme 1 suit une procédure d'apprentissage par correction d'erreur. L'analyseur doit donc savoir détecter si le choix d'une action induit une erreur. Lorsque aucune erreur n'est détectée, le vecteur de paramètres n'est pas modifié et la construction de l'arbre

de dépendances continue selon les prédictions du modèle. Par conséquent, l'algorithme 1 peut être employé *sans modification* pour entraîner un analyseur à partir de données partiellement annotées. En effet, une erreur est détectée lorsqu'une action prédite empêche la construction d'une dépendance de la structure de référence ; si aucune information de supervision n'est disponible (c.-à-d. si la dépendance correcte n'est pas connue), aucune action n'est pénalisante dans le processus de construction de la sortie et toutes les actions prédites peuvent être considérées comme correctes. Les informations non connues n'entravent donc pas l'apprentissage sur le reste de la structure connue. La procédure d'apprentissage n'a donc pas besoin d'être modifiée, c'est le calcul de l'ensemble $\textsc{Correct}_y(c)$ qui s'adapte automatiquement aux informations disponibles.

Algorithm 1: Apprentissage d'un analyseur par transitions

for $t \in [\![1, ..., T]\!]$ **do**
  $x, y \leftarrow \textsc{Sample}(\text{dataset})$ // extraction d'une phrase
  $c \leftarrow \textsc{Initial}(x)$ // configuration initiale
  while $\neg\textsc{Final}(c)$ **do**
    $a^* = \arg\max_{a \in \textsc{Legal}(c)} \langle \phi(c, a) | \mathbf{w} \rangle$ // action prédite
    if $a^* \notin \textsc{Correct}_y(c)$ **then**
      $\hat{a} = \arg\max_{a \in \textsc{Correct}_y(c)} \langle \phi(c, a) | \mathbf{w} \rangle$ // action correcte de score max.
      $\mathbf{w} \leftarrow \mathbf{w} + \phi(c, \hat{a}) - \phi(c, a^*)$ // mise à jour
    $c \leftarrow c \circ a^*$

La méthode d'apprentissage que nous venons de décrire présente un second avantage lui permettant d'exploiter les dépendances partielles. L'oracle dynamique a été introduit pour limiter l'accumulation d'erreurs lors de l'apprentissage d'un analyseur en dépendances. En effet, lors de l'apprentissage l'analyseur construit une structure de dépendances à partir d'actions prédites donc potentiellement erronées. Ce qui signifie que, au fur et à mesure de la construction de cette structure, l'apprentissage du vecteur de paramètres utilise des traits qui proviennent d'une structure prédite potentiellement fausse : l'analyseur apprend à prédire les bonnes actions dans un contexte erroné ce qui le rend mieux à même de s'adapter aux erreurs de prédiction en phase de test. Cette caractéristique est particulièrement importante dans le cadre de l'apprentissage partiel puisque l'analyseur, lors de l'apprentissage, peut se retrouver dans un état dans lequel les dépendances précédemment prédites n'ont pas été validées en comparaison avec la structure de référence (qui est inconnue) et sont donc potentiellement fausses.

Il est intéressant de noter toutefois que la détection d'une erreur est possible même lorsque la tête du mot courant n'est pas connue : dans certains cas, grâce à la contrainte de projectivité, la connaissance des dépendances voisines permet de restreindre les actions correctes possibles. La figure 1 illustre un exemple de structure partielle dans laquelle le mot « libre » (sans tête connue) a seulement deux choix de tête possibles (« le », son voisin gauche, ou « échange », son voisin droit) du fait de la dépendance connue entre « échange » et « Le » et de la contrainte de projectivité. En effet, celle-ci proscrit le rattachement d'un mot compris entre la tête et le dépendant d'une dépendance à une tête située à l'extérieur de cet intervalle. Lors de l'entraînement de l'analyseur par transition, cela se traduit par une restriction sur l'ensemble des actions correctes. Ici, après l'application de la première prédiction SHIFT (qui place « Le » sur la pile), l'ensemble des actions correctes, $\textsc{Correct}_y(c)$, est limité à SHIFT et RIGHTARC tandis que l'ensemble des actions possibles $\textsc{Legal}(c)$ contient également l'action LEFTARC. Si cette dernière action est choisie, alors il y a une mise à jour du vecteur de poids

bien que la tête du mot « libre » ne soit pas connue. Si l'une des actions SHIFT ou RIGHTARC est prédite, il n'y a pas de mise à jour à cette étape mais, à l'étape suivante, les actions correctes seront également restreintes : si SHIFT est choisie, alors « libre » n'aura plus qu'un choix de rattachement correct (« échange », par l'action LEFTARC) tandis que si RIGHTARC est choisie, alors l'action REDUCE sera le seul choix correct possible. Cette restriction peut entraîner une mise à jour.

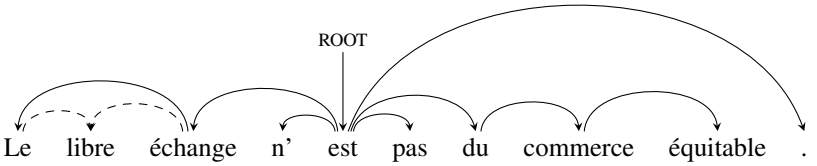


FIGURE 1 – Structure de dépendances partielle (traits pleins) dans laquelle la tête du mot « libre » n'est pas connue mais est limitée, à cause de la contrainte de projectivité à deux choix (traits en pointillé) : « Le » ou « échange ».

2.3 Expériences sur des données artificielles

Afin de montrer que l'apprentissage à partir de données partiellement annotées est possible, nous avons mené des expériences de contrôle sur des données dont les dépendances ont été supprimées artificiellement : nous comparons les performances d'un analyseur appris, d'une part, à partir de $n\%$ des phrases d'un corpus, et d'autre part, à partir de toutes les phrases du corpus mais pour lesquelles seules $n\%$ des dépendances sont connues. Dans les deux cas, les données conservées pour l'apprentissage (phrases ou dépendances) sont extraites aléatoirement.

La figure 2 présente les performances en UAS [1] des analyseurs sur les corpus allemand et espagnol du *Universal Dependency Treebank* [2] (UDT) v2.0 standard (McDonald *et al.*, 2013) selon le pourcentage de dépendances préservées durant l'entraînement. Les résultats correspondent à un score moyen mesuré sur 10 expériences pour lesquelles nous conservons la division des corpus proposée par le corpus UDT en données d'entraînement et données de test, à l'exclusion des phrases non-projectives [3].

Les UAS des analyses débutent à 75,44% pour l'allemand et à 78,83% pour l'espagnol lorsque seulement 10% des phrases (intégralement annotées) sont conservées pour l'apprentissage, et atteignent respectivement 80,35% et 82,81% sur l'intégralité du corpus. Lorsque l'apprentissage se fait sur l'intégralité des corpus auxquels sont aléatoirement retirées des dépendances, les scores sont toujours supérieurs (au maximum +0,43 pour l'allemand et +0,62 pour l'espagnol) pour un nombre moyen équivalent de dépendances conservées. Nous avons effectué des expériences similaires sur d'autres langues du UDT pour lesquelles les résultats obtenus suivent les mêmes tendances.

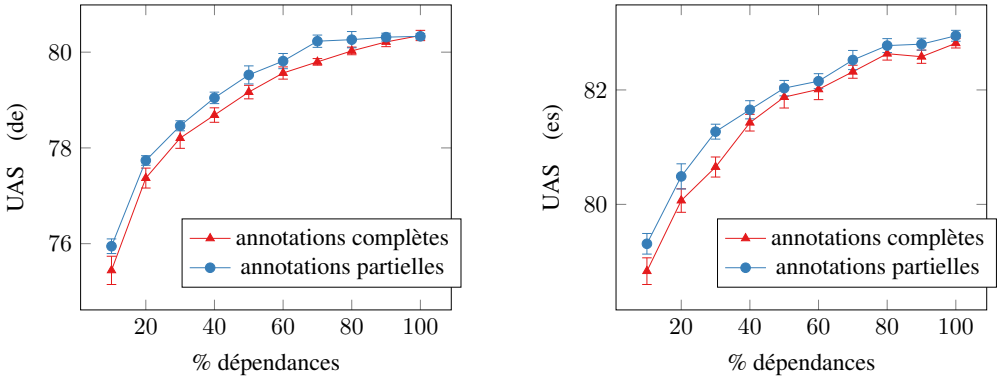
Nous remarquons donc d'une part que le nombre de phrases d'entraînement peut être considérablement réduit sans affaiblir sévèrement les scores : supprimer la moitié des phrases des corpus réduit les

1. UAS (*Unlabeled Attachment Score*) correspond au pourcentage de mots étant correctement rattachés sur l'ensemble du corpus, à l'exception des ponctuations.

2. `https://github.com/ryanmcd/uni-dep-tb`

3. Dans nos expériences, le corpus allemand contient 12 752 phrases dans le corpus d'apprentissage et 785 dans le corpus de test, tandis que les corpus espagnol comprennent, respectivement, 13 280 et 267 phrases.

scores de « seulement » 1,18 points pour l'allemand et 0,94 pour l'espagnol. Pour chacune des deux langues et pour un même nombre de dépendances connues, les scores sont supérieurs lorsque plus de phrases sont annotées, même partiellement. Comme expliqué à la section 2.2, des informations sur les dépendances inconnues peuvent en effet être déduites du contexte syntaxique (c.-à-d. les dépendances voisines connues) et ainsi corriger des erreurs lors de l'apprentissage. En pratique, lors de l'apprentissage sur 60 % des dépendances 35 382 mises à jour sont effectuées en moyenne contre 31 339 lorsque seulement 60 % des phrases sont exploitées dans le cas de l'allemand ce qui constitue une différence importante (+13% de mises à jours) bien que le nombre moyen de dépendances conservées soit équivalent. Des écarts similaires ont été relevés pour d'autres langues de l'UDT.



(a) Langue : Allemand

(b) Langue : Espagnol

FIGURE 2 – UAS d'un analyseur entraîné sur $n\%$ des dépendances des corpus UDT allemand et espagnol. « Annotations complètes » : $n\%$ des phrases (intégralement annotées) du corpus. « Annotations partielles » : $n\%$ des dépendances de tout le corpus.

3 Application au transfert de dépendances

Dans cette section, nous montrons comment l'apprentissage à partir de données partiellement annotées peut être employé pour développer des analyseurs performants pour des langues peu dotées en transférant les annotations d'une langue mieux dotée. Nous proposons dans un premier temps une méthode simple de projection partielle des dépendances d'une langue source vers une langue cible. Cette méthode permet d'obtenir des données cibles partiellement annotées en se fiant uniquement aux alignements non ambigus. Puis nous utilisons ces données pour apprendre des analyseurs en dépendances pour les langues cibles. Cette approche est validée expérimentalement en comparant, d'une part, notre méthode de transfert avec des méthodes état-de-l'art dans ce domaine et, d'autre part, en observant l'impact du choix des heuristiques de filtrage.

3.1 Transfert de dépendance par projection partielle

Les grands corpus de textes parallèles alignés phrase-à-phrase utilisés en traduction permettent de transférer des annotations d'une langue vers l'autre. Par exemple, dans le cas de la projection de

dépendances, le texte source provenant d'une langue bien dotée est automatiquement annoté, puis ces annotations sont projetées sur le texte cible via les liens d'alignement entre les mots (voir figure 3) : lorsque dans la phrase source un mot x_1 est dépendant d'un mot x_2, et que chacun de ces mots est aligné respectivement avec un unique mot, y_1 et y_2, dans la phrase cible, un lien de dépendance est établi de y_2 vers y_1. Dans la figure 3, le mot « this » est dépendant du mot « is » dans la phrase source (anglais) et ces deux mots sont respectivement alignés (uniquement) avec les mots « ce » et « est » en cible (français) ; la dépendance entre « is » et « this » peut donc être projetée en cible de « est » à « ce » avec une grande confiance.

La projection des dépendances via les liens d'alignement `1:1` est intuitivement, une façon assez sûre d'obtenir des dépendances correctes pour la phrase cible. Cette approche ne permet toutefois pas de déterminer les dépendances des mots impliqués dans des alignements multiples (de type `n:m`) et pour les mots non-alignés. Hwa *et al.* (2005) ont proposé des heuristiques pour déterminer les dépendances manquantes dans ces deux cas. Mais ces heuristiques ont pour principal objectif de construire, pour les phrases cibles, des structures de dépendances complètes, sans nécessairement tenir compte de la plausibilité de celles-ci. Au final, elles reposent sur des décisions arbitraires allant, parfois, jusqu'à l'ajout de mots « factices » dans la phrase cible. C'est pourquoi elles restent peu fiables et produisent des données de supervision très bruitées.

Dans ce travail, nous avons choisi une approche à la fois plus simple et plus robuste : nous proposons d'ignorer les alignements « difficiles » ou inexistants et de ne considérer que la projection des dépendances correspondant aux liens d'alignement `1:1`[4], tel qu'illustré par la figure 3. Cette approche permet de produire dans la langue cible des données partiellement annotées, contenant uniquement les dépendances que nous qualifions de « sûres », car elles reposent directement sur les liens d'alignement observés et non sur des décisions arbitraires. L'avantage d'une telle méthode est qu'elle permet d'obtenir des annotations de qualité très simplement, sans nécessiter une étape coûteuse de modélisation de la confiance des dépendances projetées. En outre, elle est indépendante de la langue et donc applicable à n'importe quelle paire de langues.

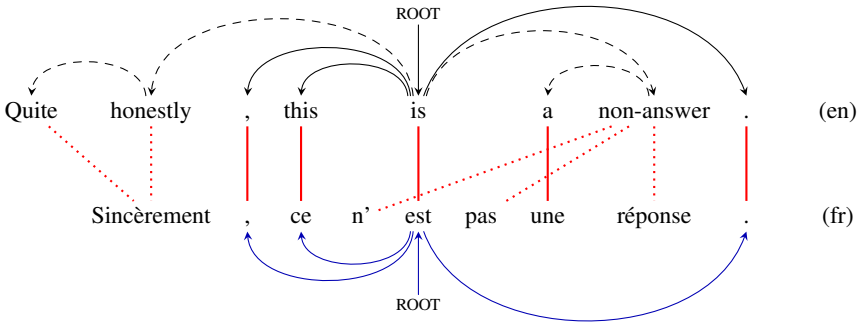


FIGURE 3 – Projection partielle de l'anglais vers le français. Seules les dépendances sources dont la tête et le dépendant sont alignés avec un et un seul mot cible sont projetées (dépendances et alignements représentés par des traits pleins). Les traits pointillés décrivent les alignements n:m.

4. Dans le cas de la dépendance attachant la racine de la phrase, nous considérons un mot additionnel dans chacune des phrases, source et cible, toujours correctement aligné.

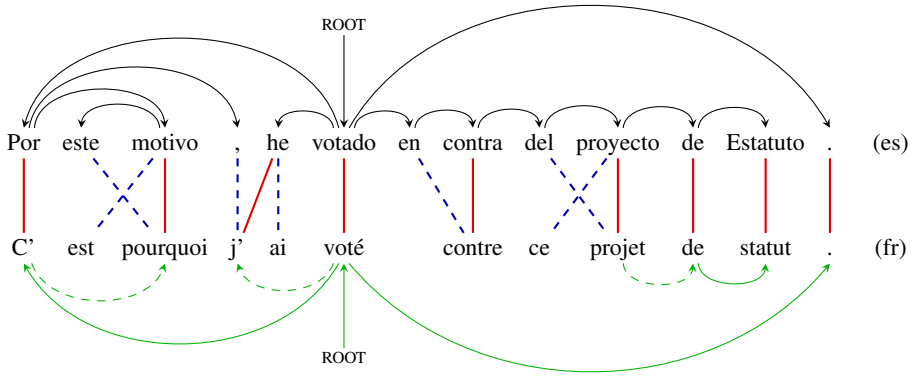


FIGURE 4 – Projection partielle de l'espagnol vers le français via les alignements en fonction de l'heuristique de symétrisation employée. Les liens d'alignement rouges (pleins) sont ceux qui apparaissent dans les deux sens d'alignement (source vers cible et cible vers source) ; seuls ces liens sont conservés lors de la symétrisation via l'heuristique *intersection*. Les liens d'alignement en bleu (pointillés) apparaissent dans une seule direction d'alignement ; ils sont conservés en plus des liens rouges (pleins) lors de la symétrisation avec l'heuristique *grow-diag*. Les dépendances vertes (en dessous) sont celles qui sont projetées avec l'emploi de *intersection* et seules les dépendances en traits pleins verts sont projetées avec *grow-diag*.

3.2 Processus de transfert

Alignements Tous les textes parallèles sont automatiquement alignés dans les deux directions [5]. Deux heuristiques sont utilisées pour fusionner les deux directions d'alignements : *intersection* et *grow-diag* (Koehn, 2010). L'heuristique *intersection* ne conserve que les liens prédits conjointement dans les deux directions, qui correspondent intuitivement aux alignements `1:1` les plus sûrs. L'heuristique *grow-diag* complète les alignements en utilisant des arguments de voisinage ; contrairement à l'heuristique *intersection*, elle génère des alignements multiples. D'une certaine manière, l'heuristique *grow-diag* apporte plus d'informations en conservant l'ambiguïté existante entre les alignements.

Analyse en dépendances Pour chaque paire de langues traitée, il est nécessaire que les données sources soient annotées en partie du discours et en dépendances (avant projection sur les données cibles). Pour cela, nous choisissons le MateParser (Bohnet & Nivre, 2012), un système état-de-l'art performant pour l'étiquetage en partie du discours joint à l'analyse en dépendances, qui est appris sur les données d'apprentissage de l'UDT, pour étiqueter et analyser les différents corpus source.

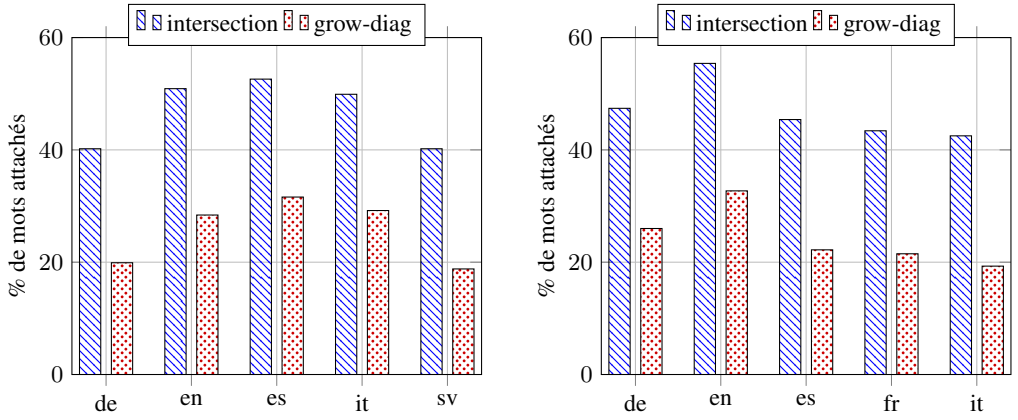
Pour l'étape d'analyse en dépendances des corpus partiellement annotés (après projection), nous employons notre propre implémentation de l'analyseur en dépendances par transition *Arc-Eager* associé à un oracle dynamique et utilisant la recherche par faisceaux *(beam-search)* avec un faisceau de taille 8. Les traits employés pour l'apprentissage sont ceux proposés par Zhang & Nivre (2011).

5. Dans nos expériences nous utilisons FASTALIGN (Dyer *et al.*, 2013) qui implémente un modèle IBM 2.

Projection partielle Les dépendances sont projetées (partiellement) sur les données cibles en utilisant la méthode décrite en 3.1 via les alignements symétrisés à l'aide des heuristiques *intersection* et *grow-diag*. La figure 4 donne un exemple des deux possibilités de projection induites par les deux cas de symétrisation. La dépendance existante entre « he » et « votado » en espagnol est projetée en français entre les tokens « j' » et « voté » avec l'emploi de *intersection* car seuls les liens d'alignements `1:1` (« he » – « j' » et « votado » – « voté ») sont conservés ; tandis qu'avec *grow-diag* certains liens `n:m` sont conservés en plus, ici « he » et « j' » sont impliqués dans des alignements multiples, ils sont donc ignorés lors de la projection.

L'heuristique *intersection* permet de projeter plus de dépendances que l'heuristique *grow-diag* bien qu'elle génère moins de liens d'alignement. En effet, comme elle ne conserve que des liens `1:1`, tous les tokens alignés peuvent recevoir une dépendance alors qu'avec l'heuristique *grow-diag* tous les liens d'alignement multiples sont ignorés lors de la projection.

Nous constatons de fait l'impact du choix de la symétrisation sur l'étape de projection partielle. Cette étape de projection partielle permet d'obtenir un taux d'attachement des mots qui va de 17,9% (suédois vers italien) à 33,1% (espagnol vers italien) avec l'heuristique *grow-diag* et de 38,2% (italien vers allemand) à 55,9% (espagnol vers anglais) avec l'heuristique *intersection*. Le nombre de dépendances transférées dans le cas de deux langues est présenté dans la figure 5. Nous détaillerons l'impact sur les performances de l'analyse dans la section 3.4.



(a) Langue cible : Français

(b) Langue cible : Suédois

FIGURE 5 – Pourcentage de mots attachés (c.-à-d. recevant une dépendance) sur l'ensemble des mots des corpus (1 231 216 phrases) après projection via les alignements *intersection* / *grow-diag*.

Filtrage Pour garantir la qualité des dépendances projetées, nous appliquons deux règles de filtrage. Premièrement, pour les projections partielles, les alignements mettant en jeu des mots en source et en cible dont les parties du discours ne font pas partie de la même classe d'équivalence (suivant les règles « *soft* » proposées par Rasooli & Collins (2015)) sont retirés [6]. Deuxièmement, nous choisissons de ne pas conserver pour l'apprentissage les phrases non-projectives car elles correspondent généralement à

6. Le filtrage par les parties du discours est nécessaire pour les phrases qui reçoivent une projection partielle pour garantir la qualité de la projection mais désavantageux lorsqu'il est appliqué sur les phrases qui reçoivent une projection complète (sans filtrage) car restreignant la projection aux structures syntaxiques similaires entre les langues source et cible.

des projections de faible qualité (Mareček, 2011). Enfin nous retirons également les phrases ayant peu de dépendances car il s'agit souvent de phrases « mal » alignées (c.-à-d. ayant peu de liens d'alignements `1:1`, et dont on peut donc supposer que l'alignement est peu fiable).

Au regard des résultats de nos expériences, nous décidons de retirer les phrases ayant moins de 80% de mots attachés. Après ce filtrage, le pourcentage de mots recevant une dépendance, sur l'ensemble des phrases, varie de 89,86% (français vers italien) à 96,12% (allemand vers italien) pour un nombre de phrases qui va de 1 583 (allemand vers italien) à 14 369 (espagnol vers italien) avec *grow-diag* ; et de 87,5% (espagnol vers français) à 91,5% (italien vers allemand) pour un nombre de phrases qui va de 4 727 (italien vers allemand) à 52 554 (anglais vers suédois) avec l'*intersection*.

3.3 Expériences

Nous effectuons nos expériences sur 6 langues [7] de l'UDT (v2.0 standard) : allemand (de), anglais (en), espagnol (es), français (fr), italien (it) et suédois (sv). En ce qui concerne les données parallèles, nous considérons un sous-ensemble du corpus Europarl (Koehn, 2005) constitué des phrases qui sont communes à chacune des langues que nous étudions, rassemblant 1 231 216 phrases parallèles pour chaque paire de langues.

Chaque ensemble de données partiellement annoté (par projection partielle) est exploité pour entraîner des analyseurs en dépendances pour chaque paire de langues et leurs performances sont évaluées sur les données de test des corpus UDT (en utilisant des étiquettes en partie du discours prédites par le MateParser). Le critère d'évaluation est le UAS (excluant la ponctuation).

Nous proposons également de comparer notre méthode avec des méthodes de transfert couramment présentées comme référence et de réaliser des expériences multi-sources pour lesquelles un ensemble de langues, plutôt qu'une seule langue, est utilisé comme une unique source.

Méthodes de l'état-de-l'art Nous comparons notre méthode de transfert de dépendances à plusieurs méthodes état-de-l'art. La première est une méthode qui inclut, en plus du parsing délexicalisé, une étape intermédiaire de re-lexicalisation sur des données de la langue cible (McDonald *et al.*, 2011). La seconde, proposée par Ma & Xia (2014), emploie une méthode de régularisation d'entropie pour transférer des connaissances cross-lingues et la dernière est la méthode « *density-driven* » de Rasooli & Collins (2015) utilisant une méthode d'analyse contrainte pour compléter les arbres partiels projetés. Cette dernière est celle qui se rapproche le plus de la méthode proposée. Nous utilisons, dans nos travaux, les mêmes corpus parallèles (Europarl) et les mêmes corpus en dépendances pour l'entrainement et l'évaluation (UDT v2.0 std) que dans ces travaux états-de-l'art. Notons toutefois que les conditions d'expérimentations n'étant pas similaires [8] à celles de notre méthode toute comparaison directe est impossible.

Transfert multi-source Pour chacune des langues cibles nous présentons également les résultats du transfert multi-source pour lequel l'ensemble des 5 autres langues est considérée comme une unique langue source. Nous proposons une stratégie qui consiste pour une phrase donnée :

7. Il s'agit des 6 langues présentes à la fois dans le corpus Europarl et le UDT.

8. Les étapes d'apprentissage (des langues sources et cibles) de notre méthode n'incluent pas certains traits tels que les clusters, et appliquent une largeur de faisceaux inférieure.

1. à réaliser l'intersection entre les structures projetées depuis les 5 langues dans le but d'obtenir une structure partielle commune aux 5 langues : si un mot est attaché à la même tête dans toutes les structures alors cet attachement est conservé ;
2. à ajouter à cette structure les dépendances les plus fréquentes apparaissant dans les structures projetées (et par ordre de fréquence), si celles-ci garantissent toujours la propriété d'arbre de la structure obtenue.

Cette approche peut engendrer des structures de dépendances partielles, si les seules possibilités de rattachement de certains tokens ne garantissent pas la propriété d'arbre de la structure.

3.4 Résultats

Les performances des différentes méthodes de transfert sont présentées dans le tableau 2. Dans un premier temps, nous nous concentrons sur les résultats obtenus lorsque le transfert est opéré à partir de l'anglais comme langue source, qui est le cas de transfert le plus étudié dans la littérature, du fait des bonnes performances des analyseurs pour l'anglais et de la qualité des alignements. Dans ce cas, les performances en UAS que nous obtenons sont systématiquement supérieures avec l'emploi de l'heuristique de symétrisation *intersection* qu'avec l'heuristique *grow-diag* (de +1,14 (fr) à +2,49 (de)). Ces scores sont cohérents avec la quantité de données transférées : la qualité des dépendances prédites est d'autant plus élevée que le nombre de phrases considérées lors de l'apprentissage (après filtrage) est élevé. Avec un même seuil de filtrage des phrases ayant peu de dépendances, les ensembles d'apprentissage induits par l'emploi de l'heuristique *grow-diag* comprennent moins d'annotations et donc probablement moins de diversité syntaxique.

En nous concentrant sur les scores obtenus à l'aide de l'heuristique *intersection*, nous constatons que la méthode que nous proposons obtient des résultats UAS significativement supérieurs à la méthode de re-lexicalisation de McDonald *et al.* (2011) (de +3,98 (de) à +8,15 (es)). Nos scores surpassent ceux de Ma & Xia (2014) pour 4 langues (+1,34 (es), +1,39 (fr), +2,85 (sv), +0,08(it)). Nous atteignons des scores équivalents à Rasooli & Collins (2015) pour le suédois mais inférieurs pour les autres langues.

		M11	**MX14**	**RC15**		**ces travaux**				**sup.**
						intersection		*grow-diag*		
source		(en)	(en)	(en)	(multi)	(en)	(multi)	(en)	(multi)	
cible	**de**	69,77	74,30	74,32	79,68	73,75	76,87	71,26	75,62	84,43
	es	68,72	75,53	78,17	80,86	76,87	79,31	75,46	79,01	85,51
	fr	73,13	76,53	79,91	82,72	77,92	80,91	76,78	81,06	85,81
	it	70,74	77,74	79,46	83,67	77,82	80,10	76,25	80,14	86,97
	sv	75,87	79,27	82,11	84,06	82,12	83,34	80,16	83,14	87,89

TABLE 2 – Résultats des évaluations de notre méthode de transfert et comparaison avec des méthodes récentes de l'état-de-l'art : M11 correspond à (McDonald *et al.*, 2011), MX14 à (Ma & Xia, 2014), RC15 à (Rasooli & Collins, 2015). Ces scores proviennent de l'article de Rasooli & Collins (2015). 'sup' présente les scores obtenus par apprentissage complètement supervisé.

Ces derniers écarts de score s'expliquent entre autres par des conditions d'apprentissage différentes (voir section 3.3) : en effet Rasooli & Collins (2015) indiquent des scores supervisés en moyenne 1,03 supérieurs aux nôtres. Lorsque nous tentons de reproduire les résultats qu'ils atteignent lors de la première étape de leur processus (i.e. apprentissage effectué uniquement sur les structures

intégralement projetées) nous obtenons des scores effectivement inférieurs de 1,5 en moyenne. Cependant, la méthode que nous proposons permet de gagner 3,3 points en moyenne sur ces scores, ce qui témoigne de l'intérêt de l'apprentissage sur les données partiellement annotées. [9]

Dans le cas du transfert multi-source, nous obtenons des résultats significativement supérieurs au transfert mono-source. Notons que, en multi-source, l'écart entre les résultats obtenus par les heuristiques *intersection* et *grow-diag* est fortement réduit, et parfois même, les scores sont quasiment équivalents avec *grow-diag* (+0.15 (fr) et +0.04 (it)). La diversité syntaxique dégradée en mono-source semble être recouverte par l'union des informations transférées depuis différentes langues.

Il est important de remarquer que la méthode que nous proposons atteint des scores capables de concurrencer ces méthodes état-de-l'art récentes tout en étant beaucoup plus simple à mettre en œuvre et en étant algorithmiquement beaucoup moins coûteuse : nos résultats sont obtenus par l'entraînement d'un seul analyseur avec une largeur de faisceaux de 8 alors que Rasooli & Collins (2015) nécessite 4 entraînements (et d'analyse avec contraintes) avec une largeur de faisceaux de 64, et Ma & Xia (2014) emploie un analyseur avec inférence exacte de complexité $\mathcal{O}(n^4)$.

Notons également que les scores des différentes méthodes de transfert restent très éloignés des scores d'un analyseur appris de manière supervisé. Des expériences supplémentaires sont nécessaires pour déterminer si cette baisse est due à une limite intrinsèque des méthodes de transfert (les langues n'étant pas isomorphes certaines constructions syntaxiques ne seront jamais vues) ou aux données utilisées lors de l'apprentissage (contrairement à l'apprentissage supervisé, l'évaluation des méthodes de transfert se fait sur un corpus hors-domaine).

4 Conclusion

Nous avons montré qu'il est possible d'apprendre un analyseur en dépendances à partir de données partiellement annotées grâce à un oracle dynamique. En utilisant des annotations partielles artificiellement produites, nous montrons alors que, pour un même nombre de dépendances, il est préférable de conserver pour l'apprentissage un grand ensemble de phrases partiellement annotées plutôt qu'un ensemble de phrases restreint intégralement annoté. Cette observation nous a permis d'utiliser cette méthode d'apprentissage partiel dans le cadre du transfert d'annotations syntaxiques entre langues. Nous avons privilégié une méthode simple de projection des dépendances entre les langues dans le but de projeter uniquement les dépendances que nous estimons les plus « sûres » quand bien même les structures de dépendances obtenues soient incomplètes. Avec un choix approprié de l'heuristique de symétrisation et du filtrage, ce processus s'est avéré très efficace pour apprendre rapidement des analyseurs pour les langues cibles et bien moins coûteux que les alternatives récentes.

Remerciements

Ces travaux ont été en partie financés par un projet DGA-RAPID à travers la subvention N° 1429060465 (Papyrus). Nous remercions les relecteurs pour leurs commentaires et suggestions.

9. Nos scores sont également supérieurs à ceux obtenus par Rasooli & Collins (2015) lors de leur première étape. Notre méthode n'étant pas seulement concurrente mais également complémentaire à la leur, elle pourrait être substituée à la première étape de leur processus dans le but d'améliorer leurs scores finaux.

Références

BOHNET B. & NIVRE J. (2012). A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, p. 1455–1465, Jeju Island, Korea.

COHEN S. B., DAS D. & SMITH N. A. (2011). Unsupervised Structure Prediction with Non-Parallel Multilingual Guidance. In *Proceedings of EMNLP 2011, the Conference on Empirical Methods in Natural Language Processing*, p. 50–61, Edinburgh, Scotland, UK.

DYER C., CHAHUNEAU V. & SMITH N. A. (2013). A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of NAACL 2013,the Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 644–648, Atlanta, Georgia : Association for Computational Linguistics.

GOLDBERG Y. & NIVRE J. (2012). A Dynamic Oracle for Arc-Eager Dependency Parsing. In *Proceedings of COLING 2012, the International Conference on Computational Linguistics*, p. 959–976, Bombay, India.

HWA R., RESNIK P., A.WEINBERG, CABEZAS C. & KOLAK O. (2005). Bootstrapping Parsers via Syntactic Projection accross Parallel Texts. *Natural language engineering*, **11**, 311–325.

KOEHN P. (2005). Europarl : A Parallel Corpus for Statistical Machine Translation. In *Proceedings of MT Summit X*, p. 79–86, Phuket, Thailand.

KOEHN P. (2010). *Statistical Machine Translation*. New York, NY, USA : Cambridge University Press, 1st edition.

LI Z., ZHANG M. & CHEN W. (2014). Soft Cross-lingual Syntax Projection for Dependency Parsing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics : Technical Papers*, p. 783–793, Dublin, Ireland : Dublin City University and Association for Computational Linguistics.

MA X. & XIA F. (2014). Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1337–1348, Baltimore, Maryland.

MAREČEK D. (2011). Combining Diverse Word-Alignment Symmetrizations Improves Dependency Tree Projection. In *Computational Linguistics and Intelligent Text Processing*, p. 144–154. Springer.

MCDONALD R., NIVRE J., QUIRMBACH-BRUNDAGE Y., GOLDBERG Y., DAS D., GANCHEV K., HALL K., PETROV S., ZHANG H., TÄCKSTRÖM O., BEDINI C., BERTOMEU CASTELLÓ N. & LEE J. (2013). Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of ACL 2013, the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 92–97, Sofia, Bulgaria.

MCDONALD R., PETROV S. & HALL K. (2011). Multi-source Transfer of Delexicalized Dependency Parsers. In *Proceedings of EMNLP 2011, the Conference on Empirical Methods in Natural Language Processing*, p. 62–72.

NASEEM T., CHEN H., BARZILAY R. & JOHNSON M. (2010). Using Universal Linguistic Knowledge to Guide Grammar Induction. In *Proceedings of EMNLP 2010, the Conference on Empirical Methods in Natural Language Processing*, p. 1234–1244, Stroudsburg, PA, USA.

NIVRE J. (2003). An Efficient Algorithm for Projective Dependency Parsing. In *Proceedings of IWPT 2003, the 8th International Workshop on Parsing Technologies*, Nancy, France.

OZDOWSKA S. (2006). Projecting POS Tags and Syntactic Dependencies from English and French to Polish in Aligned Corpora. In *Proceedings of CrossLangInduction 2006, the International Workshop on Cross-Language Knowledge Induction*, p. 53–60, Stroudsburg, PA, USA : Association for Computational Linguistics.

PAN S. J. & YANG Q. (2010). A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, **22**(10), 1345–1359.

PETROV S., DAS D. & MCDONALD R. (2012). A universal part-of-speech tagset. In N. C. C. CHAIR), K. CHOUKRI, T. DECLERCK, M. U. DOĞAN, B. MAEGAARD, J. MARIANI, J. ODIJK & S. PIPERIDIS, Eds., *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey : European Language Resources Association (ELRA).

RASOOLI M. S. & COLLINS M. (2015). Density-driven cross-lingual transfer of dependency parsers. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 328–338, Lisbon, Portugal : Association for Computational Linguistics.

SØGAARD A. (2011). Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of ACL 2011, the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, p. 682–686, Portland, Oregon, USA.

SPREYER K. & KUHN J. (2009). Data-Driven Dependency Parsing of New Languages Using Incomplete and Noisy Training Data. In *Proceedings of CoNLL 2009, the Thirteenth Conference on Computational Natural Language Learning*, p. 12–20, Boulder, Colorado.

TÄCKSTRÖM O., MCDONALD R. & NIVRE J. (2013). Target Language Adaptation of Discriminative Transfer Parsers. In *Proceedings of ACL 2013, the Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1061–1071, Atlanta, Georgia.

TIEDEMANN J. (2014). Rediscovering Annotation Projection for Cross-Lingual Parser Induction. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics : Technical Papers*, p. 1854–1864, Dublin, Ireland : Dublin City University and Association for Computational Linguistics.

TIEDEMANN J., AGIĆ V. & NIVRE J. (2014). Treebank translation for cross-lingual parser induction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, p. 130–140, Ann Arbor, Michigan : Association for Computational Linguistics.

ZEMAN D. & RESNIK P. (2008). Cross-Language Parser Adaptation between Related Languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, p. 35–42, Hyderabad, India : Asian Federation of Natural Language Processing.

ZHANG Y. & NIVRE J. (2011). Transition-based Dependency Parsing with Rich Non-local Features. In *Proceedings of ACL 2011, the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, p. 188–193, Portland, Oregon, USA : Association for Computational Linguistics.