Probabilistic Lexical Generalization for French Dependency Parsing

Enrique Henestroza Anguiano and Marie Candito
Alpage (Université Paris Diderot / INRIA)
Paris, France
enrique.henestroza_anguiano@inria.fr, marie.candito@linguist.jussieu.fr

Abstract

This paper investigates the impact on French dependency parsing of lexical generalization methods beyond lemmatization and morphological analysis. A distributional thesaurus is created from a large text corpus and used for distributional clustering and WordNet automatic sense ranking. The standard approach for lexical generalization in parsing is to map a word to a single generalized class, either replacing the word with the class or adding a new feature for the class. We use a richer framework that allows for probabilistic generalization, with a word represented as a probability distribution over a space of generalized classes: lemmas, clusters, or synsets. Probabilistic lexical information is introduced into parser feature vectors by modifying the weights of lexical features. We obtain improvements in parsing accuracy with some lexical generalization configurations in experiments run on the French Treebank and two out-of-domain treebanks, with slightly better performance for the probabilistic lexical generalization approach compared to the standard single-mapping approach.

1 Introduction

In statistical, data-driven approaches to natural language syntactic parsing, a central problem is that of accurately modeling lexical relationships from potentially sparse counts within a training corpus. Our particular interests are centered on reducing lexical data sparseness for linear classification approaches for dependency parsing. In these approaches, linear models operate over feature vectors that generally represent syntactic structure within a sentence, and feature templates are defined in part over the word forms of one or more tokens in a sentence. Because treebanks used for training are often small, lexical features may appear relatively infrequently during training, especially for languages with richer morphology than English. This may, in turn, impede the parsing model's ability to generalize well outside of its training set with respect to lexical features.

Past approaches for achieving lexical generalization in dependency parsing have used WordNet semantic senses in parsing experiments for English (Agirre et al., 2011), and word clustering over large corpora in parsing experiments for English (Koo et al., 2008) as well as for French (Candito et al., 2010b). These approaches map each word to a single corresponding generalized class (synset or cluster), and integrate generalized classes into parsing models in one of two ways: (i) the *replacement strategy*, where each word form is simply replaced with a corresponding generalized class; (ii) a strategy where an additional feature is created for the corresponding generalized class.

Our contribution in this paper is applying *probabilistic lexical generalization*, a richer framework for lexical generalization, to dependency parsing. Each word form is represented as a categorical distribution over a *lexical target space* of generalized classes, for which we consider the spaces of lemmas, synsets, and clusters. The standard single-mapping approach from previous work can be seen as a subcase: each categorical distribution assigns a probability of 1 to a single generalized class. The method

we use for introducing probabilistic information into a feature vector is based on that used by Bunescu (2008), who tested the use of probabilistic part-of-speech (POS) tags through an NLP pipeline.

In this paper, we perform experiments for French that use the replacement strategy for integrating generalized classes into parsing models, comparing the single-mapping approach for lexical generalization with our probabilistic lexical generalization approach. In doing so, we provide first results on the application to French parsing of WordNet automatic sense ranking (ASR), using the method of McCarthy et al. (2004). For clustering we deviate from most previous work, which has integrated Brown clusters (Brown et al., 1992) into parsing models, and instead use distributional lexical semantics to create both a distributional thesaurus - for probabilistic generalization in the lemma space and ASR calculation - and to perform hierarchical agglomerative clustering (HAC). Though unlexicalized syntactic HAC clustering has been used to improve English dependency parsing (Sagae and Gordon, 2009), we provide first results on using distributional lexical semantics for French parsing. We also include an out-of-domain evaluation on medical and parliamentary text in addition to an in-domain evaluation.

In Section 2 we describe the lexical target spaces used in this paper, as well as the method of integrating probabilistic lexical information into a feature vector for classification. In Section 3 we discuss dependency structure and transition-based parsing. In Section 4 we present the experimental setup, which includes our parser implementation, the construction of our probabilistic lexical resources, and evaluation settings. We report parsing results both in-domain and out-of-domain in Section 5, we provide a summary of related work in Section 6, and we conclude in Section 7.

2 Probabilistic Lexical Target Spaces

Using terms from probability theory, we define a *lexical target space* as a sample space Ω over which a categorical distribution is defined for each lexical item in a given *source vocabulary*. Because we are working with French, a language with relatively rich morphology, we use lemmas as the base lexical items in our source vocabulary. The outcomes contained in a sample space represent generalized classes in a *target vocabulary*. In this paper we consider three possible target vocabularies, with corresponding sample spaces: Ω_l for lemmas, Ω_s for synsets, and Ω_c for clusters.

2.1 Ω_l Lemma Space

In the case of the lemma space, the source and target vocabularies are the same. To define an appropriate categorical distribution for each lemma, one where the possible outcomes also correspond to lemmas, we use a *distributional thesaurus* that provides similarity scores for pairs of lemmas. Such a thesaurus can be viewed as a similarity function $D(x, y)$, where $x, y \in V$ and V is the vocabulary for both the source and target spaces.

The simplest way to define a categorical distribution over Ω_l, for a lemma $x \in V$, would be to use the following probability mass function p_x:

$$p_x(y) = \frac{D(x, y)}{\sum_{y' \in V} D(x, y')} \quad (1)$$

One complication is the identity similarity $D(x, x)$: although it can be set equal to 1 (or the similarity given by the thesaurus, if one is provided), we choose to assign a pre-specified probability mass m to the identity lemma, with the remaining mass used for generalization across other lemmas. Additionally, in order to account for noise in the thesaurus, we restrict each categorical distribution to a lemma's k-nearest neighbors. The probability mass function p_x over the space Ω_l that we use in this paper is finally as follows:

$$p_x(y) = \begin{cases} m, & \text{if } y = x \\ \frac{(1-m)D(x,y)}{\sum_{y' \in N_x(k)} D(x, y')}, & \text{if } y \in N_x(k) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

2.2 Ω_s Synset Space

In the case of the synset space, the target vacabulary contains synsets from the Princeton WordNet sense hierarchy (Fellbaum, 1998). To define an appropriate categorical distribution over synsets for each

lemma x in our source vocabulary, we first use the WordNet resource to identify the set S_x of different senses of x. We then use a distributional thesaurus to perform ASR, which determines the prevalence with respect to x of each sense $s \in S_x$, following the approach of McCarthy et al. (2004). Representing the thesaurus as a similarity function $D(x, y)$, letting $N_x(k)$ be the set of k-nearest neighbors for x, and letting $W(s_1, s_2)$ be a similarity function over synsets in WordNet, we define a prevalence function $R_x(s)$ as follows:

$$R_x(s) = \sum_{y \in N_x(k)} D(x,y) \frac{\max_{s' \in S_y} W(s,s')}{\sum_{t \in S_x} \max_{s' \in S_y} W(t,s')} \quad (3)$$

This function essentially weights the semantic contribution that each distributionally-similar neighbor adds to a given sense for x. With the prevalence scores of each sense for x having been calculated, we use the following probability mass function p_x over the space Ω_s, where $S_x(k)$ is the set of k-most prevalent senses for x:

$$p_x(s) = \begin{cases} \dfrac{R_x(s)}{\sum_{s' \in S_x(k)} R_x(s')}, & \text{if } s \in S_x(k) \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Note that the first-sense ASR approach to using WordNet synsets for parsing, which has been previously explored in the literature (Agirre et al., 2011), corresponds to setting k=1 in Equation 4.

2.3 Ω_c Cluster Space

In the case of the cluster space, any approach for word clustering may be used to create a reduced target vocabulary of clusters. Defining a categorical distribution over clusters would be interesting in the case of *soft clustering* of lemmas, in which a lemma can participate in more than one cluster, but we have not yet explored this clustering approach.

In this paper we limit ourselves to the simpler *hard clustering* HAC method, which uses a distributional thesaurus and iteratively joins two clusters together based on the similarities between lemmas in each cluster. We end up with a simple probability mass function p_x over the space Ω_c for a lemma x with corresponding cluster c_x:

$$p_x(c) = \begin{cases} 1, & \text{if } c = c_x \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

2.4 Probabilistic Feature Generalization

In a typical classifier-based machine learning setting in NLP, feature vectors are constructed using indicator functions that encode categorical information, such as POS tags, word forms or lemmas.

In this section we will use a running example where a and b are token positions of interest to a classifier, and for which feature vectors are created. If we let t stand for POS tag and l stand for lemma, a *feature template* for this pair of tokens might then be $[t_a l_b]$. Feature templates are instantiated as actual features in a vector space depending on the categorical values they can take on. One possible instantiation of the template $[t_a l_b]$ would then be the feature $[t_a{=}verb \wedge l_b{=}avocat]$, which indicates that a is a verb and b is the lemma $avocat$ ("avocado" or "lawyer"), with the following indicator function:

$$f = \begin{cases} 1, & \text{if } t_a{=}verb \wedge l_b{=}avocat \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

To perform probabilistic feature generalization, we replace the indicator function, which represents a single original feature, with a collection of weighted functions representing a set of derived features. Suppose the French lemma $avocat$ is in our source vocabulary and has multiple senses in Ω_s (s_1 for the "avocado" sense, s_2 for the "lawyer" sense, etc.), as well as a probability mass function p_{av}. We discard the old feature $[t_a{=}verb \wedge l_b{=}avocat]$ and add, for each s_i, a derived feature of the form $[t_a{=}verb \wedge x_b{=}s_i]$, where x represents a target space generalized class, with the following weighted indicator function:

$$f(i) = \begin{cases} p_{av}(s_i), & \text{if } t_a{=}verb \wedge l_b{=}avocat \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

This process extends easily to generalizing multiple categorical variables. Consider the bilexical feature $[l_a{=}manger \wedge l_b{=}avocat]$, which indicates that a is the lemma $manger$ ("eat") and b is the lemma $avocat$. If both lemmas $manger$ and $avocat$ appear

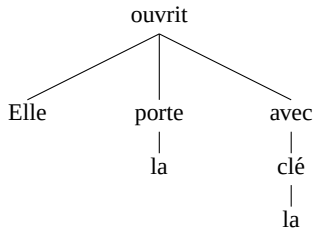

Figure 1: An unlabeled dependency tree for "Elle ouvrit la porte avec la clé" ("She opened the door with the key").

in our source vocabulary and have multiple senses in Ω_s, with probability mass functions p_{ma} and p_{av}, then for each pair i, j we derive a feature of the form [x_a=$s_i \wedge x_b$=s_j], with the following weighted indicator function:

$$f(i,j) = \begin{cases} p_{ma}(s_i)p_{av}(s_j), & \text{if } l_a = manger \wedge l_b = avocat \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

3 Dependency Parsing

Dependency syntax involves the representation of syntactic information for a sentence in the form of a directed graph, whose edges encode word-to-word relationships. An edge from a *governor* to a *dependent* indicates, roughly, that the presence of the dependent is syntactically legitimated by the governor. An important property of dependency syntax is that each word, except for the root of the sentence, has exactly one governor; dependency syntax is thus represented by trees. Figure 1 shows an example of an unlabeled dependency tree.[1] For languages like English or French, most sentences can be represented with a *projective* dependency tree: for any edge from word g to word d, g dominates any intervening word between g and d.

Dependency trees are appealing syntactic representations, closer than constituency trees to the semantic representations useful for NLP applications. This is true even with the projectivity requirement, which occasionally creates syntax-semantics mismatches. Dependency trees have recently seen a surge of interest, particularly with the introduction of supervised models for dependency parsing using linear classifiers.

[1]Our experiments involve labeled parsing, with edges additionally labeled with the surface grammatical function that the dependent bears with respect to its governor.

3.1 Transition-Based Parsing

In this paper we focus on transition-based parsing, whose seminal works are that of Yamada and Matsumoto (2003) and Nivre (2003). The parsing process applies a sequence of incremental actions, which typically manipulate a buffer position in the sentence and a stack for built sub-structures. In the *arc-eager* approach introduced by Nivre et al. (2006) the possible actions are as follows, with s_0 being the token on top of the stack and n_0 being the next token in the buffer:

- SHIFT: Push n_0 onto the stack.
- REDUCE: Pop s_0 from the stack.
- RIGHT-ARC(r): Add an arc labeled r from s_0 to n_0; push n_0 onto the stack.
- LEFT-ARC(r): Add an arc labeled r from n_0 to s_0; pop s_0 from the stack.

The parser uses a greedy approach, where the action selected at each step is the best-scoring action according to a classifier, which is trained on a dependency treebank converted into sequences of actions. The major strength of this framework is its $O(n)$ time complexity, which allows for very fast parsing when compared to more complex global optimization approaches.

4 Experimental Setup

We now discuss the treebanks used for training and evaluation, the parser implementation and baseline settings, the construction of the probabilistic lexical resources, and the parameter tuning and evaluation settings.

4.1 Treebanks

The treebank we use for training and in-domain evaluation is the French Treebank (FTB) (Abeillé and Barrier, 2004), consisting of 12,351 sentences from the *Le Monde* newspaper, converted to projective[2] dependency trees (Candito et al., 2010a). For our experiments we use the usual split of 9,881 training, 1,235 development, and 1,235 test sentences.

[2]The projectivity constraint is linguistically valid for most French parses: the authors report $< 2\%$ non-projective edges in a hand-corrected subset of the converted FTB.

Moving beyond the journalistic domain, we use two additional treebank resources for out-of-domain parsing evaluations. These treebanks are part of the Sequoia corpus (Candito and Seddah, 2012), and consist of text from two non-journalistic domains annotated using the FTB annotation scheme: a medical domain treebank containing 574 development and 544 test sentences of public assessment reports of medicine from the European Medicines Agency (EMEA) originally collected in the OPUS project (Tiedemann, 2009), and a parliamentary domain treebank containing 561 test sentences from the Europarl[3] corpus.

4.2 Parser and Baseline Settings

We use our own Python implementation of the arc-eager algorithm for transition-based parsing, based on the arc-eager setting of MaltParser (Nivre et al., 2007), and we train using the standard FTB training set. Our baseline feature templates and general settings correspond to those obtained in a benchmarking of parsers for French (Candito et al., 2010b), under the setting which combined lemmas and morphological features.[4] Automatic POS-tagging is performed using MElt (Denis and Sagot, 2009), and lemmatization and morphological analysis are performed using the Le*fff* lexicon (Sagot, 2010). Table 1 lists our baseline parser's feature templates.

4.3 Lexical Resource Construction

We now describe the construction of our probabilistic lexical target space resources, whose prerequisites include the automatic parsing of a large corpus, the construction of a distributional thesaurus, the use of ASR on WordNet synsets, and the use of HAC clustering.

4.3.1 Automatically-Parsed Corpus

The text corpus we use consists of 125 million words from the *L'Est Republicain* newspaper[5], 125 million words of dispatches from the *Agence France-Presse*, and 225 million words from a French Wikipedia backup dump[6]. The corpus is preprocessed using the Bonsai tool[7], and parsed using our baseline parser.

	Feature Templates
Unigram	$t_{n_0};\ l_{n_0};\ c_{n_0};\ w_{n_0};\ t_{s_0};\ l_{s_0};\ c_{s_0}; w_{s_0};\ d_{s_0};$ $t_{n_1};\ l_{n_1};\ t_{n_2};\ t_{n_3};\ t_{s_1};\ t_{s_2};\ t_{n_0l};\ l_{n_0l};\ d_{n_0l};$ $d_{s_{0l}};\ d_{s_{0r}};\ l_{s_{0h}};\ \{m^i_{n_0} : i \in \lvert M\rvert\};$ $\{m^i_{s_0} : i \in \lvert M\rvert\}$
Bigram	$t_{s_0}t_{n_0};\ t_{s_0}l_{n_0};\ l_{s_0}l_{n_0};\ l_{n_0}t_{n_1};\ t_{n_0}t_{n_0l};$ $t_{n_0}d_{n_0l};\ \{m^i_{s_0}m^j_{n_0} : i; j \in \lvert M\rvert\};$ $\{t_{n_0}m^i_{n_0} : i \in \lvert M\rvert\}; \{t_{s_0}m^i_{s_0} : i \in \lvert M\rvert\}$
Trigram	$t_{s_2}t_{s_1}t_{s_0};\ t_{s_1}t_{s_0}t_{n_0};\ t_{s_0}t_{n_0}t_{n_1};\ t_{n_0}t_{n_1}t_{n_2};$ $t_{n_1}t_{n_2}t_{n_3};\ t_{s_0}d_{s_{0l}}d_{s_{0r}}$

Table 1: Arc-eager parser feature templates. c = coarse POS tag, t = fine POS tag, w = inflected word form, l = lemma, d = dependency label, m^i = morphological feature from set M. For tokens, $n_i = i^{th}$ token in the buffer, $s_i = i^{th}$ token on the stack. The token subscripts l, r, and h denote partially-constructed syntactic left-most dependent, right-most dependent, and head, respectively.

4.3.2 Distributional Thesaurus

We build separate distributional thesauri for nouns and for verbs,[8] using straightforward methods in distributional lexical semantics based primarily on work by Lin (1998) and Curran (2004). We use the FreDist tool (Henestroza Anguiano and Denis, 2011) for thesaurus creation.

First, *syntactic contexts* for each lemma are extracted from the corpus. We use all syntactic dependencies in which the secondary token has an open-class POS tag, with labels included in the contexts and two-edge dependencies used in the case of prepositional-phrase attachment and coordination. Example contexts are shown in Figure 2. For verb lemmas we limit contexts to dependencies in which the verb is governor, and we add unlexicalized versions of contexts to account for subcategorization. For noun lemmas, we use all dependencies in which the noun participates, and all contexts are lexicalized. The vocabulary is limited to lemmas with at least 1,000 context occurrences, resulting in 8,171 nouns and 2,865 verbs.

Each pair of lemma x and context c is subsequently weighted by mutual informativeness using the point-wise mutual information metric, with

[3] http://www.statmt.org/europarl/

[4] That work tested the use of Brown clusters, but obtained no improvement compared to a setting without clusters. Thus, we do not evaluate Brown clustering in this paper.

[5] http://www.cnrtl.fr/corpus/estrepublicain/

[6] http://dumps.wikimedia.org/

[7] http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html

[8] We additionally considered adjectives and adverbs, but our initial tests yielded no parsing improvements.

· One-Edge Context: $-obj\rightarrow N|avocat$

· One-Edge Context: (unlexicalized) $-obj\rightarrow N$

· Two-Edge Context: $-mod\rightarrow P|avec\ -obj\rightarrow N|avocat$

· Two-Edge Context: (unlexicalized) $-mod\rightarrow P|avec\ -obj\rightarrow N$

Figure 2: Example dependency contexts for the verb lemma $manger$. The one-edge contexts corresponds to the phrase "manger un avocat" ("eat an avocado"), and the two-edge contexts corresponds to the phrase "manger avec un avocat" ("eat with a lawyer").

probabilities estimated using frequency counts:

$$I(x,c) = log\left(\frac{p(x,c)}{p(x)p(c)}\right) \quad (9)$$

Finally, we use the cosine metric to calculate the distributional similarity between pairs of lemmas x, y:

$$D(x,y) = \frac{\sum_c I(x,c)I(y,c)}{\sqrt{\left(\sum_c I(x,c)^2\right) \times \left(\sum_c I(y,c)^2\right)}} \quad (10)$$

4.3.3 WordNet ASR

For WordNet synset experiments we use the French EuroWordNet[9] (FREWN). A WordNet synset mapping[10] allows us to convert synsets in the FREWN to Princeton WordNet version 3.0, and after discarding a small number of synsets that are not covered by the mapping we retain entries for 9,833 nouns and 2,220 verbs. We use NLTK, the Natural Language Toolkit (Bird et al., 2009), to calculate similarity between synsets. As explained in Section 2.2, ASR is performed using the method of McCarthy et al. (2004). We use k=8 for the distributional nearest-neighbors to consider when ranking the senses for a lemma, and we use the synset similarity function of Jiang and Conrath (1997), with default information content counts from NLTK calculated over the British National Corpus[11].

[9] http://www.illc.uva.nl/EuroWordNet/

[10] http://nlp.lsi.upc.edu/tools/download-map.php

[11] http://www.natcorp.ox.ac.uk/

	Source Vocabulary	Evaluation Set		
		FTB Eval	EMEA Eval	Europarl
Nouns	FTB train	95.35	62.87	94.69
	Thesaurus	96.25	79.00	97.83
	FREWN	80.51	73.09	87.06
Verbs	FTB train	96.54	94.56	97.76
	Thesaurus	98.33	97.82	99.54
	FREWN	88.32	91.48	91.98

Table 2: Lexical occurrence coverage (%) of source vocabularies over evaluation sets. FTB Eval contains both the FTB development and test sets, while EMEA Eval contains both the EMEA development and test sets. Proper nouns are excluded from the analysis.

4.3.4 HAC Clustering

For the HAC clustering experiments in this paper, we use the CLUTO package[12]. The distributional thesauri described above are taken as input, and the UPGMA setting is used for cluster agglomeration. We test varying levels of clustering, with a parameter z which determines the proportion of cluster vocabulary size with respect to the original vocabulary size (8,171 for nouns and 2,865 for verbs).

4.3.5 Resource Coverage

The coverage of our lexical resources over the FTB and two out-of-domain evaluation sets, at the level of token occurrences of verbs and common nouns, is described in Table 2. We can see that the FTB training set vocabulary provides better coverage than the FREWN for both nouns and verbs, while the coverage of the thesauri (and derived clusters) is the highest overall.

4.4 Tuning and Evaluation

We evaluate four lexical target space configurations against the baseline of lemmatization, tuning parameters using ten-fold cross-validation on the FTB training set. The feature templates are the same as those in Table 1, with the difference that features involving lemmas are modified by the probabilistic feature generalization technique described in Section 2.4, using the appropriate categorical distributions. In all configurations, we exclude the French auxiliary verbs *être* and *avoir* from participation in lexical generalization, and we replace proper nouns

[12] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download

with a special lemma[13]. Below we describe the tuned parameters for each configuration.

- **RC: Replacement with cluster in Ω_c**
  For clusters and the parameter z (cf. Section 4.3.4), we settled on relative cluster vocabulary size z=0.6 for nouns and z=0.7 for verbs. We also generalized lemmas not appearing in the distributional thesaurus into a single unknown class.

- **PKNL: Probabilistic k-nearest lemmas in Ω_l**
  For the parameters k and m (cf. Section 2.1), we settled on k=4 and m=0.5 for both nouns and verbs. We also use the unknown class for low-frequency lemmas, as in the RC configuration.

- **RS: Replacement with first-sense (k=1) in Ω_s**
  Since the FREWN has a lower-coverage vocabulary, we did not use an unknown class for out-of-vocabulary lemmas; instead, we mapped them to unique senses. In addition, we did not perform lexical generalization for verbs, due to low cross-validation performance.

- **PKPS: Probabilistic k-prevalent senses in Ω_s**
  For this setting we decided to not place any limit on k, due to the large variation in the number of senses for different lemmas. As in the RS configuration, we mapped out-of-vocabulary lemmas to unique senses and did not perform lexical generalization for verbs.

5 Results

Table 3 shows labeled attachment score (LAS) results for our baseline parser (Lemmas) and four lexical generalization configurations. For comparison, we also include results for a setting that only uses word forms (Forms), which was the baseline for previous work on French dependency parsing (Candito et al., 2010b). Punctuation tokens are not scored, and significance is calculated using Dan Bikel's randomized parsing evaluation comparator[14], at significance level p=0.05.

[13]Proper nouns tend to have sparse counts, but for computational reasons we did not include them in our distributional thesaurus construction. We thus chose to simply generalize them into a single class.

[14]http://www.cis.upenn.edu/~dbikel/software.html

Parse Configuration	Evaluation Set LAS			
	FTB Test	EMEA Dev	EMEA Test	Europarl
Forms	86.85	84.08	85.41	86.01
Lemmas	87.30	84.34	85.41	86.26
RC	87.32	84.28	85.71*	86.28
PKNL	87.46	84.63*	85.82*	86.26
RS	87.34	84.48	85.54	86.34
PKPS	87.41	84.63*	85.68*	86.22

Table 3: Labeled attachment score (LAS) on in-domain (FTB) and out-of-domain (EMEA, Europarl) evaluation sets for the baseline (Lemmas) and four lexical generalization configurations (RC, PKNL, RS, PKPS). Significant improvements over the baseline are starred. For comparison, we also include a simpler setting (Forms), which does not use lemmas or morphological features.

5.1 In-Domain Results

Our in-domain evaluation yields slight improvements in LAS for some lexical generalization configurations, with PKNL performing the best. However, the improvements are not statistically significant. A potential explanation for this disappointing result is that the FTB training set vocabulary covers the FTB test set at high rates for both nouns (95.25%) and verbs (96.54%), meaning that lexical data sparseness is perhaps not a big problem for in-domain dependency parsing. While WordNet synsets could be expected to provide the added benefit of taking word sense into account, sense ambiguity is not really treated due to ASR not providing word sense disambiguation in context.

5.2 Out-Of-Domain Results

Our evaluation on the medical domain yields statistically significant improvements in LAS, particularly for the two probabilistic target space approaches. PKNL and PKPS improve parsing for both the EMEA dev and test sets, while RC improves parsing for only the EMEA test set and RS does not significantly improve parsing for either set. As in our in-domain evaluation, PKNL performs the best overall, though not significantly better than other lexical generalization settings. One explanation for the improvement in the medical domain is the substantial increase in coverage of nouns in EMEA afforded

by the distributional thesaurus (+26%) and FREWN (+16%) over the base coverage afforded by the FTB training set.

Our evaluation on the parliamentary domain yields no improvement in LAS across the different lexical generalization configurations. Interestingly, Candito and Seddah (2012) note that while Europarl is rather different from FTB in its syntax, its vocabulary is surprisingly similar. From Table 2 we can see that the FTB training set vocabulary has about the same high level of coverage over Europarl (94.69% for nouns and 97.76% for verbs) as it does over the FTB evaluation sets (95.35% for nouns and 96.54% for verbs). Thus, we can use the same reasoning as in our in-domain evaluation to explain the lack of improvement for lexical generalization methods in the parliamentary domain.

5.3 Lexical Feature Use During Parsing

Since lexical generalization modifies the lexical feature space in different ways, we also provide an analysis of the extent to which each parsing model's lexical features are used during in-domain and out-of-domain parsing. Table 4 describes, for each configuration, the number of lexical features stored in the parsing model along with the *average lexical feature use* (ALFU) of classification instances (each instance represents a parse transition) during training and parsing.[15]

Lexical feature use naturally decreases when moving from the training set to the evaluation sets, due to holes in lexical coverage outside of a parsing model's training set. The single-mapping configurations (RC, RS) do not increase the number of lexical features in a classification instance, which explains the fact that their ALFU on the FTB training set (6.0) is the same as that of the baseline. However, the decrease in ALFU when parsing the evaluation sets is less severe for these configurations than for the baseline: when parsing EMEA Dev with the RC configuration, where we obtain a significant LAS improvement over the baseline, the reduction in ALFU is only 13% compared to 22% for the baseline parser. For the probabilistic generalization configurations, we also see decreases in ALFU when parsing the evaluation sets, though their higher absolute ALFU may help explain the strong medical domain parsing performance for these configurations.

[15] We define the lexical feature use of a classification instance to be the number of lexical features in the parsing model that receive non-zero values in the instance's feature vector.

Parse Configuration	Lexical Feats In Model	Average Lexical Feature Use		
		FTB Train	FTB Dev	EMEA Dev
Lemmas	294k	6.0	5.5	4.7
RC	150k	6.0	5.8	5.2
PKNL	853k	15.7	14.8	12.0
RS	253k	6.0	5.6	4.9
PKPS	500k	9.2	8.6	7.0

Table 4: Parsing model lexical features (rounded to nearest thousand) and average lexical feature use in classification instances across different training and evaluation sets, for the baseline (Lemmas) and four lexical generalization configurations (PKNL, RC, PKPS, and RS).

5.4 Impact on Running Time

Another factor to note when evaluating lexical generalization is the effect that it has on running time. Compared to the baseline, the single-mapping configurations (RC, RS) speed up feature extraction and prediction time, due to reduced dimensionality of the feature space. On the other hand, the probabilistic generalization configurations (PKNL, PKPS) slow down feature extraction and prediction time, due to an increased dimensionality of the feature space and a higher ALFU. Running time is therefore a factor that favors the single-mapping approach over our proposed probabilistic approach.

Taking a larger view on our findings, we hypothesize that in order for lexical generalization to improve parsing, an approach needs to achieve two objectives: (i) generalize sufficiently to ensure that lemmas not appearing in the training set are nonetheless associated with lexical features in the learned parsing model; (ii) substantially increase lexical coverage over what the training set can provide. The first of these objectives seems to be fulfilled through our lexical generalization methods, as indicated in Table 4. The second objective, however, seems difficult to attain when parsing text in-domain, or even out-of-domain if the domains have a high lexical overlap (as is the case for Europarl). Only for our parsing experiments in the medical domain do both objectives appear to be fulfilled, as evidenced by our LAS improvements when parsing EMEA with lexical generalization.

6 Related Work

We now discuss previous work concerning the use of lexical generalization for parsing, both in the classic in-domain setting and in the more recently popular out-of-domain setting.

6.1 Results in Constituency-Based Parsing

The use of word classes for parsing dates back to the first works on generative constituency-based parsing, whether using semantic classes obtained from hand-built resources or less-informed classes created automatically. Bikel (2000) tried incorporating WordNet-based word sense disambiguation into a parser, but failed to obtain an improvement. Xiong et al. (2005) generalized bilexical dependencies in a generative parsing model using Chinese semantic resources (CiLin and HowNet), obtaining improvements for Chinese parsing. More recently, Agirre et al. (2008) show that replacing words with WordNet semantic classes improves English generative parsing. Lin et al. (2009) use the HowNet resource within the split-merge PCFG framework (Petrov et al., 2006) for Chinese parsing: they use the first-sense heuristic to append the most general hypernym to the POS of a token, obtaining a semantically-informed symbol refinement, and then guide further symbol splits using the HowNet hierarchy. Other work has used less-informed classes, notably unsupervised word clusters. Candito and Crabbé (2009) use Brown clusters to replace words in a generative PCFG-LA framework, obtaining substantial parsing improvements for French.

6.2 Results in Dependency Parsing

In dependency parsing, word classes are integrated as features in underlying linear models. In a seminal work, Koo et al. (2008) use Brown clusters as features in a graph-based parser, improving parsing for both English and Czech. However, attempts to use this technique for French have lead to no improvement when compared to the use of lemmatization and morphological analysis (Candito et al., 2010b). Sagae and Gordon (2009) augment a transition-based English parser with clusters using unlexicalized syntactic distributional similarity: each word is represented as a vector of counts of emanating unlexicalized syntactic paths, with counts taken from a corpus of auto-parsed phrase-structure trees, and HAC clustering is performed using cosine similarity. For semantic word classes, (Agirre et al., 2011) integrate WordNet senses into a transition-based parser for English, reporting small but significant improvements in LAS (+0.26% with synsets and +0.36% with semantic files) on the full Penn Treebank with first-sense information from Semcor.

We build on previous work by attempting to reproduce, for French, past improvements for in-domain English dependency parsing with generalized lexical classes. Unfortunately, our results for French do not replicate the improvements for English using semantic sense information (Agirre et al., 2011) or word clustering (Sagae and Gordon, 2009). The primary difference between our paper and previous work, though, is our evaluation of a novel probabilistic approach for lexical generalization.

6.3 Out-Of-Domain Parsing

Concerning techniques for improving out-of-domain parsing, a related approach has been to use self-training with auto-parsed out-of-domain data, as McClosky and Charniak (2008) do for English constituency parsing, though in that approach lexical generalization is not explicitly performed. Candito et al. (2011) use word clustering for domain adaptation of a PCFG-LA parser for French, deriving clusters from a corpus containing text from both the *source* and *target* domains, and they obtain parsing improvements in both domains. We are not aware of previous work on the use of lexical generalization for improving out-of-domain dependency parsing.

7 Conclusion

We have investigated the use of probabilistic lexical target spaces for reducing lexical data sparseness in a transition-based dependency parser for French. We built a distributional thesaurus from an automatically-parsed large text corpus, using it to generate word clusters and perform WordNet ASR. We tested a standard approach to lexical generalization for parsing that has been previously explored, where a word is mapped to a single cluster or synset. We also introduced a novel probabilistic lexical generalization approach, where a lemma

is represented by a categorical distribution over the space of lemmas, clusters, or synsets. Probabilities for the lemma space were calculated using the distributional thesaurus, and probabilities for the WordNet synset space were calculated using ASR sense prevalence scores, with probabilistic clusters left for future work.

Our experiments with an arc-eager transition-based dependency parser resulted in modest but significant improvements in LAS over the baseline when parsing out-of-domain medical text. However, we did not see statistically significant improvements over the baseline when parsing in-domain text or out-of-domain parliamentary text. An explanation for this result is that the French Treebank training set vocabulary has a very high lexical coverage over the evaluation sets in these domains, suggesting that lexical generalization does not provide much additional benefit. Comparing the standard single-mapping approach to the probabilistic generalization approach, we found a slightly (though not significantly) better performance for probabilistic generalization across different parsing configurations and evaluation sets. However, the probabilistic approach also has the downside of a slower running time.

Based on the findings in this paper, our focus for future work on lexical generalization for dependency parsing is to continue improving parsing performance on out-of-domain text, specifically for those domains where lexical variation is high with respect to the training set. One possibility is to experiment with building a distributional thesaurus that uses text from both the source and target domains, similar to what Candito et al. (2011) did with Brown clustering, which may lead to a stronger *bridging* effect across domains for probabilistic lexical generalization methods.

Acknowledgments

This work was funded in part by the ANR project Sequoia ANR-08-EMER-013.

References

A. Abeillé and N. Barrier. 2004. Enriching a French treebank. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal, May.

E. Agirre, T. Baldwin, and D. Martinez. 2008. Improving parsing and PP attachment performance with sense information. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 317–325, Columbus, Ohio, June.

E. Agirre, K. Bengoetxea, K. Gojenola, and J. Nivre. 2011. Improving dependency parsing with semantic classes. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 699–703, Portland, Oregon, June.

D.M. Bikel. 2000. A statistical model for parsing and word-sense disambiguation. In *Proceedings of the EMNLP/VLC-2000*, pages 155–163, Hong Kong, October.

S. Bird, E. Loper, and E. Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

P.F. Brown, P.V. Desouza, R.L. Mercer, V.J.D. Pietra, and J.C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

R.C. Bunescu. 2008. Learning with probabilistic features for improved pipeline models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 670–679, Honolulu, Hawaii, October.

M. Candito and B. Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *Proceedings of the 11th International Conference on Parsing Technologies*, pages 138–141, Paris, France, October.

M. Candito and D. Seddah. 2012. Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *Actes de la 19ème conférence sur le traitement automatique des langues naturelles*, Grenoble, France, June. To Appear.

M. Candito, B. Crabbé, and P. Denis. 2010a. Statistical French dependency parsing: Treebank conversion and first results. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valetta, Malta, May.

M. Candito, J. Nivre, P. Denis, and E. Henestroza Anguiano. 2010b. Benchmarking of statistical dependency parsers for French. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 108–116, Beijing, China, August.

M. Candito, E. Henestroza Anguiano, D. Seddah, et al. 2011. A Word Clustering Approach to Domain Adaptation: Effective Parsing of Biomedical Texts. In *Proceedings of the 12th International Conference on Parsing Technologies*, Dublin, Ireland, October.

J.R. Curran. 2004. *From distributional to semantic similarity*. Ph.D. thesis, University of Edinburgh.

P. Denis and B. Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art

POS tagging with less human effort. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, Hong Kong, China, December.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

E. Henestroza Anguiano and P. Denis. 2011. FreDist: Automatic construction of distributional thesauri for French. In *Actes de la 18ème conférence sur le traitement automatique des langues naturelles*, pages 119–124, Montpellier, France, June.

J.J. Jiang and D.W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference on Research in Computational Linguistics*, Taiwan.

T. Koo, X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 595–603, Columbus, Ohio, June.

X. Lin, Y. Fan, M. Zhang, X. Wu, and H. Chi. 2009. Refining grammars for parsing with hierarchical semantic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1298–1307, Singapore, August.

D. Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 768–774, Montreal, Quebec, August.

D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 279–286, Barcelona, Spain, July.

D. McClosky and E. Charniak. 2008. Self-training for biomedical parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 101–104, Columbus, Ohio, June.

J. Nivre, J. Hall, J. Nilsson, G. Eryi it, and S. Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 221–225, New York City, NY, June.

J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02):95–135.

J. Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies*, pages 149–160, Nancy, France, April.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July.

K. Sagae and A. Gordon. 2009. Clustering words by syntactic similarity improves dependency parsing of predicate-argument structures. In *Proceedings of the 11th International Conference on Parsing Technologies*, pages 192–201, Paris, France, October.

B. Sagot. 2010. The Lefff, a freely available, accurate and large-coverage lexicon for French. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valetta, Malta, May.

J. Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume 5, pages 237–248. John Benjamins, Amsterdam.

D. Xiong, S. Li, Q. Liu, S. Lin, and Y. Qian. 2005. Parsing the penn chinese treebank with semantic knowledge. In *Proceedings of the International Joint Conference on Natural Language Processing*, pages 70–81, Jeju Island, Korea, October.

H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies*, pages 195–206, Nancy, France, April.