Syntactic Transfer Using a Bilingual Lexicon

Greg Durrett, Adam Pauls, and **Dan Klein**
Computer Science Division
University of California, Berkeley
{gdurrett,adpauls,klein}@cs.berkeley.edu

Abstract

We consider the problem of using a bilingual dictionary to transfer lexico-syntactic information from a resource-rich source language to a resource-poor target language. In contrast to past work that used bitexts to transfer analyses of specific sentences at the token level, we instead use features to transfer the behavior of words at a type level. In a discriminative dependency parsing framework, our approach produces gains across a range of target languages, using two different low-resource training methodologies (one weakly supervised and one indirectly supervised) and two different dictionary sources (one manually constructed and one automatically constructed).

1 Introduction

Building a high-performing parser for a language with no existing treebank is still an open problem. Methods that use no supervision at all (Klein and Manning, 2004) or small amounts of manual supervision (Haghighi and Klein, 2006; Cohen and Smith, 2009; Naseem et al., 2010; Berg-Kirkpatrick and Klein, 2010) have been extensively studied, but still do not perform well enough to be deployed in practice. Projection of dependency links across aligned bitexts (Hwa et al., 2005; Ganchev et al., 2009; Smith and Eisner, 2009) gives better performance, but crucially depends on the existence of large, in-domain bitexts. A more generally applicable class of methods exploits the notion of universal part of speech tags (Petrov et al., 2011; Das and Petrov, 2011) to train parsers that can run on any language with no adaptation (McDonald et al., 2011) or unsupervised adaptation (Cohen et al., 2011). While these universal parsers currently constitute the highest-performing methods for languages without treebanks, they are inherently limited by operating at the coarse POS level, as lexical features are vital to supervised parsing models.

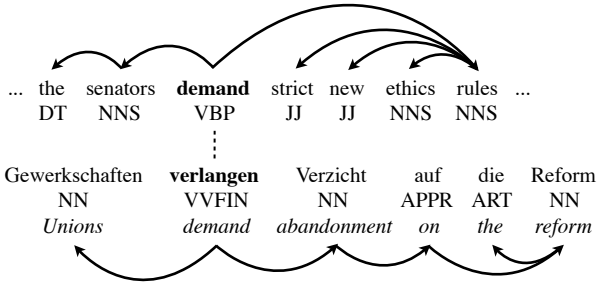

Figure 1: Sentences in English and German both containing words that mean "demand." The fact that the English *demand* takes nouns on its left and right indicates that the German *verlangen* should do the same, correctly suggesting attachments to *Verzicht* and *Gewerkschaften*.

In this work, we consider augmenting delexicalized parsers by transferring syntactic information through a bilingual lexicon at the word type level. These parsers are delexicalized in the sense that, although they receive target language words as input, their feature sets do not include indicators on those words. This setting is appropriate when there is too little target language data to learn lexical features directly. Our main approach is to add features which are lexical in the sense that they compute a function of specific target language words, but are still un-

lexical in the sense that all lexical knowledge comes from the bilingual lexicon and training data in the source language.

Consider the example English and German sentences shown in Figure 1, and suppose that we wish to parse the German side without access to a German treebank. A delexicalized parser operating at the part of speech level does not have sufficient information to make the correct decision about, for example, the choice of subcategorization frame for the verb *verlangen*. However, *demand*, a possible English translation of *verlangen*, takes a noun on its left and a noun on its right, an observation that in this case gives us the information we need. We can fire features in our German parser on the attachments of *Gewerkschaften* and *Verzicht* to *verlangen* indicating that similar-looking attachments are attested in English for an English translation of *verlangen*. This allows us to exploit fine-grained lexical cues to make German parsing decisions even when we have little or no supervised German data; moreover, this syntactic transfer is possible even in spite of the fact that *demand* and *verlangen* are not observed in parallel context.

Using type-level transfer through a dictionary in this way allows us to decouple the lexico-syntactic projection from the data conditions under which we are learning the parser. After computing feature values using source language resources and a bilingual lexicon, our model can be trained very simply using any appropriate training method for a supervised parser. Furthermore, because the transfer mechanism is just a set of features over word types, we are free to derive our bilingual lexicon either from bitext or from a manually-constructed dictionary, making our method strictly more general than those of McDonald et al. (2011) or Täckström et al. (2012), who rely centrally on bitext. This flexibility is potentially useful for resource-poor languages, where a human-curated bilingual lexicon may be broader in coverage or more robust to noise than a small, domain-limited bitext. Of course, it is an empirical question whether transferring type level information about word behavior is effective; we show that, indeed, this method compares favorably with other transfer mechanisms used in past work.

The actual syntactic information that we transfer consists of purely monolingual lexical attachment statistics computed on an annotated source language resource.[1] While the idea of using large-scale summary statistics as parser features has been considered previously (Koo et al., 2008; Bansal and Klein, 2011; Zhou et al., 2011), doing so in a projection setting is novel and forces us to design features suitable for projection through a bilingual lexicon. Our features must also be flexible enough to provide benefit even in the presence of cross-lingual syntactic differences and noise introduced by the bilingual dictionary.

Under two different training conditions and with two different varieties of bilingual lexicons, we show that our method of lexico-syntactic projection does indeed improve the performance of parsers that would otherwise be agnostic to lexical information. In all settings, we see statistically significant gains for a range of languages, with our method providing up to 3% absolute improvement in unlabeled attachment score (UAS) and 11% relative error reduction.

2 Model

The projected lexical features that we propose in this work are based on lexicalized versions of features found in MSTParser (McDonald et al., 2005), an edge-factored discriminative parser. We take MSTParser to be our underlying parsing model and use it as a testbed on which to evaluate the effectiveness of our method for various data conditions.[2] By instantiating the basic MSTParser features over coarse parts of speech, we construct a state-of-the-art delexicalized parser in the style of McDonald et al. (2011), where feature weights can be directly transferred from a source language or languages to a desired target language. When we add projected lexical features on top of this baseline parser, we do so in a way that does not sacrifice this generality: while our new features take on values that are language-specific, they interact with the model at a language-independent level. We therefore have the best of

[1] Throughout this work, we will use English as the source language, but it is possible to use any language for which the appropriate bilingual lexicons and treebanks exist. One might expect to find the best performance from using a source language closely related to the target.

[2] We train MSTParser using the included implementation of MIRA (Crammer and Singer, 2001) and use projective decoding for all experiments described in this paper.

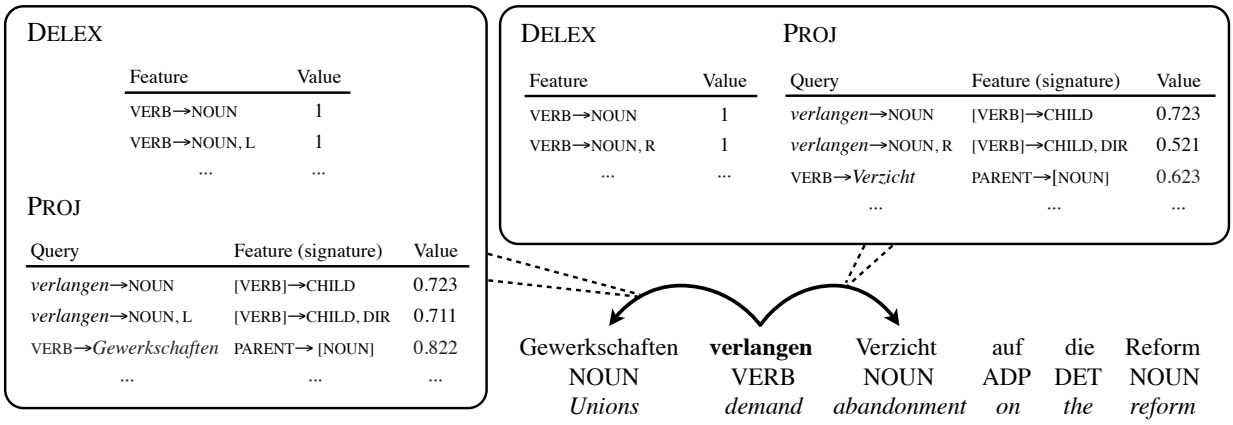

Figure 2: Computation of features on a dependency arc. DELEX features are indicators over characteristics of dependency links that do not involve the words in the sentence. PROJ features are real-valued analogues of DELEX features that do contain words. We form a query from each stipulated set of characteristics, compute the values of these queries heuristically, and then fire a feature based on each query's signature. Signatures indicate which attachment properties were considered, which part of the query was lexicalized (shown by brackets here), and the POS of the query word. This procedure yields a small number of real-valued features that still capture rich lexico-syntactic information.

two worlds in that our features can be learned on any treebank or treebanks that are available to us, but still exploit highly specific lexical information to achieve performance gains over using coarse POS features alone.

2.1 DELEX Features

Our DELEX feature set consists of all of the unlexicalized features in MSTParser, only lightly modified to improve performance for our setting. McDonald et al. (2005) present three basic types of such features, ATTACH, INBETWEEN, and SURROUNDING, which we apply at the coarse POS level. The ATTACH features for a given dependency link consist of indicators of the tags of the head and modifier, separately as well as together. The INBETWEEN and SURROUNDING features are indicators on the tags of the head and modifier in addition to each intervening tag in turn (INBETWEEN) or various combinations of tags adjacent to the head or modifier (SURROUNDING).[3]

MSTParser by default also includes a copy of each of these indicator features conjoined with the direction and distance of the attachment it denotes. These extra features are important to getting good performance out of the baseline model. We slightly modify the conjunction scheme and expand it with additional backed-off conjunctions, since these changes lead to features that empirically transfer better than the MSTParser defaults. Specifically, we use conjunctions with attachment direction (left or right), coarsened distance,[4] and attachment direction and coarsened distance combined.

We emphasize again that these baseline features are entirely standard, and all the DELEX feature set does is recreate an MSTParser-based analogue of the direct transfer parser described by McDonald et al. (2011).

2.2 PROJ Features

We will now describe how to compute our projected lexical features, the PROJ feature set, which constitutes the main contribution of this work. Recall that we wish our method to be as general as possible and work under many different training conditions; in particular, we wish to be able to train our model on only existing treebanks in other languages when no target language trees are available (discussed in Section 3.3), or on only a very small target language treebank (Section 3.4). It would greatly increase the power of our model if we were able to include target-language-lexicalized versions of the ATTACH

[3] As in Koo et al. (2008), our feature set contains more backed-off versions of the SURROUNDING features than are described in McDonald et al. (2005).

[4] Our five distance buckets are $\{1, 2, 3-5, 6-10, 11+\}$.

features, but these are not learnable without a large target language treebank. We instead must augment our baseline model with a relatively small number of features that are nonetheless rich enough to transfer the necessary lexical information.

Our overall approach is sketched in Figure 2, where we show the features that fire on two proposed edges in a German dependency parse. Features on an edge in MSTParser incorporate a subset of observable properties about that edge's head, modifier, and context in the sentence. For sets of properties that do not include a lexical item, such as VERB→NOUN, we fire an indicator feature from the DELEX feature set. For those that do include a lexical item, such as *verlangen*→NOUN, we form a *query*, which resembles a lexicalized indicator feature. Rather than firing the query as an indicator feature directly, which would result in a model parameter for each target word, we fire a broad feature called an *signature* whose value reflects the specifics of the query (computation of these values is discussed in Section 2.2.2). For example, we abstract *verlangen*→NOUN to [VERB]→CHILD, with square brackets indicating the element that was lexicalized. Section 2.2.1 discusses this coarsening in more detail. The signatures are agnostic to individual words and even the language being parsed, so they can be learned on small amounts of data or data from other languages.

Our signatures allow us to instantiate features at different levels of granularity corresponding to the levels of granularity in the DELEX feature set. When a small amount of target language data is present, the variety of signatures available to us means that we can learn language-specific transfer characteristics: for example, nouns tend to follow prepositions in both French and English, but the ordering of adjectives with respect to nouns is different. We also have the capability to train on languages other than our target language, and while this is expected to be less effective, it can still teach us to exploit some syntactic properties, such as similar verb attachment configurations if we train on a group of SVO languages distinct from a target SVO language. Therefore, our feature set manages to provide the training procedure with choices about how much syntactic information to transfer at the same time as it prevents overfitting and provides language independence.

2.2.1 Query and Signature Types

A query is a subset of the following pieces of information about an edge: parent word, parent POS, child word, child POS, attachment direction, and binned attachment distance. It must contain exactly one word.[5] We experimented with properties from INBETWEEN and SURROUNDING features as well, but found that these only helped under some circumstances and could lead to overfitting.[6]

A signature contains the following three pieces of information:

1. The non-empty subset of attachment properties included in the query
2. Whether we have lexicalized on the parent or child of the attachment, indicated by brackets
3. The part of speech of the included word

Because either the parent or child POS is included in the signature, there are three meaningful properties to potentially condition on, of which we must select a nonempty subset. Some multiplication shows that we have $7 \times 2 \times 13 = 182$ total PROJ features.

As an example, the queries

verlangen → NOUN
verlangen → ADP
sprechen → NOUN

all share the signature [VERB]→CHILD, but

verlangen → NOUN,RIGHT
Verzicht → ADP
VERB → *Verzicht*

have [VERB]→CHILD,DIR, [ADP]→CHILD, and PARENT→[NOUN] as their signatures, respectively.

The level of granularity for signatures is a parameter that simply must be engineered. We found some benefit in actually instantiating two signatures for every query, one as described above and one that

[5] Bilexical features are possible in our framework, but we do not use them here, so for clarity we assume that each query has one associated word.

[6] One hypothesis is that features looking at the sentence context are more highly specialized to a given language, since they examine the parent, the child, *and* one or more other parts of speech or words.

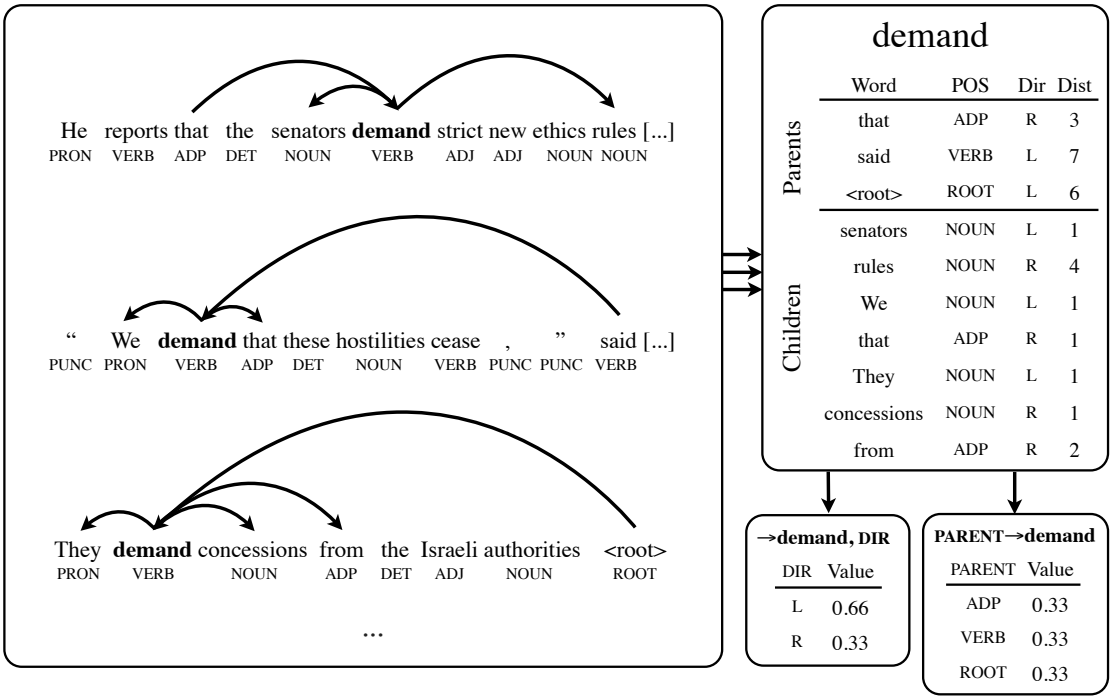

Figure 3: Computation of query values. For each occurrence of a given source word, we tabulate the attachments it takes part in (parents and children) and record their properties. We then compute relative frequency counts for each possible query type to get source language scores, which will later be projected through the dictionary to obtain target language feature values. Only two query types are shown here, but values are computed for many others as well.

does not condition on the part of speech of the word in the signature. One can also imagine using more refined signatures, but we found that this led to overfitting in the small training scenarios under consideration.

2.2.2 Query Value Estimation

Each query is given a value according to a generative heuristic that involves the source training data and the probabilistic bilingual lexicon.[7] For a particular signature, a query can be written as a tuple $(x_1, x_2, \ldots, w_t)$ where w_t is the target language query word and the x_i are the values of the included language-independent attachment properties. The value this feature takes is given by a simple generative model: we imagine generating the attachment properties x_i given w_t by first generating a source word w_s from w_t based on the bilingual lexicon, then jointly generating the x_i conditioned on w_s. Treating the choice of source translation as a latent variable to be marginalized out, we have

$$\begin{aligned}\text{value} &= p(x_1, x_2, \ldots | w_t) \\ &= \sum_{w_s} p(w_s|w_t) p(x_1, x_2, \ldots | w_s)\end{aligned}$$

The first term of the sum comes directly from our probabilistic lexicon, and the second we can estimate using the maximum likelihood estimator over our source language training data:

$$p(x_1, x_2, \ldots | w_s) = \frac{c(x_1, x_2, \ldots, w_s)}{c(w_s)} \qquad (1)$$

where $c(\cdot)$ denotes the count of an event in the source language data.

The final feature value is actually the logarithm of this computed value, with a small constant added before the logarithm is taken to avoid zeroes.

[7] Lexicons such as those produced by automatic aligners include probabilities natively, but obviously human-created lexicons do not. For these dictionaries, we simply assume that each word translates with uniform probability into each of its possible translations. Tweaking this method did not substantially change performance.

3 Experiments

3.1 Data Conditions

Before we describe the details of our experiments, we sketch the data conditions under which we evaluate our method. As described in Section 1, there is a continuum of lightly supervised parsing methods from those that make no assumptions (beyond what is directly encoded in the model), to those that use a small set of syntactic universals, to those that use treebanks from resource-rich languages, and finally to those that use both existing treebanks and bitexts.

Our focus is on parsing when one does not have access to a full-scale target language treebank, but one does have access to realistic auxiliary resources. The first variable we consider is whether we have access to a small number of target language trees or only pre-existing treebanks in a number of other languages; while not our actual target language, these other treebanks can still serve as a kind of proxy for learning which features generally transfer useful information (McDonald et al., 2011). We notate these conditions with the following shorthand:

BANKS: Large treebanks in other target languages

SEED: Small treebank in the right target language

Previous work on essentially unsupervised methods has investigated using a small number of target language trees (Smith and Eisner, 2009), but the behavior of supervised models under these conditions has not been extensively studied. We will see in Section 3.4 that with only 100 labeled trees, even our baseline model can achieve performance equal to or better than that of the model of McDonald et al. (2011). A single linguist could plausibly annotate such a number of trees in a short amount of time for a language of interest, so we believe that this is an important setting in which to show improvement, even for a method primarily intended to augment unsupervised parsing.

In addition, we consider two different sources for our bilingual lexicon:

AUTOMATIC: Extracted from bitext

MANUAL: Constructed from human annotations

Both bitexts and human-curated bilingual dictionaries are more widely available than complete treebanks. Bitexts can provide rich information about lexical correspondences in terms of how words are used in practice, but for resource-poor languages, parallel text may only be available in small quantities, or be domain-limited. We show results of our method on bilingual dictionaries derived from both sources, in order to show that it is applicable under a variety of data conditions and can successfully take advantage of such resources as are available.

3.2 Datasets

We evaluate our method on a range of languages taken from the CoNLL shared tasks on multilingual dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007). We make use of dependency treebanks for Danish, German, Greek, Spanish, Italian, Dutch, Portuguese, and Swedish, all from the 2006 shared task.

For our English resource, we use 500,000 English newswire sentences from English Gigaword version 3 (Graff et al., 2007), parsed with the Berkeley Parser (Petrov et al., 2006) and converted to a dependency treebank using the head rules of Collins (1999).[8] Our English test set (used in Section 3.4) consists of the first 300 sentences of section 23 of the Penn treebank (Marcus et al., 1993), preprocessed in the same way. Our model does not use gold fine-grained POS tags, but we do use coarse POS tags deterministically generated from the provided gold fine-grained tags in the style of Berg-Kirkpatrick and Klein (2010) using the mappings of Petrov et al. (2011).[9] Following McDonald et al. (2011), we strip punctuation from all treebanks for the results of Section 3.3. All results are given in terms of unlabeled attachment score (UAS), ignoring punctuation even when it is present.

We use the Europarl parallel corpus (Koehn, 2005) as the bitext from which to extract the AUTOMATIC bilingual lexicons. For each target language, we produce one-to-one alignments on the English-target bitext by running the Berkeley Aligner (Liang et al., 2006) with five iterations of IBM Model 1 and

[8] Results do not degrade much if one simply uses Sections 2-21 of the Penn treebank instead. Coverage of rare words in the treebank is less important when a given word must also appear in the bilingual lexicon as the translation of an observed German word in order to be useful.

[9] Note that even in the absence of gold annotation, such tags could be produced from bitext using the method of (Das and Petrov, 2011) or could be read off from a bilingual lexicon.

	This work					Past work			
		MANUAL		AUTOMATIC		MPH11*		TMU12**	
	DELEX	DELEX+PROJ	Δ	DELEX+PROJ	Δ	Multi-dir	Multi-proj Δ	No clusters	X-lingual Δ
DA	41.3	43.0	1.67 ‡	43.6	2.30 ‡	48.9*	0.6*	36.7**	2.0**
DE	58.5	58.7	0.20	59.5	0.94 †	56.7*	-0.1*	48.9**	1.8**
EL	57.9	59.9	1.99 ‡	60.5	2.55 ‡	60.1*	5.0*	59.5**	3.5**
ES	64.2	65.4	1.20 ‡	65.7	1.52 ‡	64.2*	0.3*	60.2**	2.7**
IT	65.9	66.5	0.58	67.4	1.54 ‡	64.1*	0.9*	64.6**	4.2**
NL	57.0	57.5	0.52	58.8	1.88 ‡	55.8*	9.9*	52.8**	1.5**
PT	75.4	77.2	1.83 ‡	78.7	3.29 ‡	74.0*	1.6*	66.8**	4.2**
SV	64.5	66.1	1.61 ‡	66.9	2.34 ‡	65.3*	2.7*	55.4**	1.5**
AVG	60.6	61.8	1.20	62.6	2.05	61.1*	2.7*	55.6**	2.7**

Table 1: Evaluation of features derived from AUTOMATIC and MANUAL bilingual lexicons when trained on a concatenation of non-target-language treebanks (the BANKS setting). Values reported are UAS for sentences of all lengths in the standard CoNLL test sets, with punctuation removed from training and test sets. Daggers indicate statistical significance computed using bootstrap resampling; a single dagger indicates $p < 0.1$ and a double dagger indicates $p < 0.05$. We also include the baseline results of McDonald et al. (2011) and Täckström et al. (2012) and improvements from their best methods of using bitext and lexical information. These results are not directly comparable to ours, as indicated by * and **. However, we still see that the performance of our type-level transfer method approaches that of bitext-based methods, which require complex bilingual training for each new language.

five iterations of the HMM aligner with agreement training. Our lexicon is then read off based on relative frequency counts of aligned instances of each word in the bitext.

We also use our method on bilingual dictionaries constructed in a more conventional way. For this purpose, we scrape our MANUAL bilingual lexicons from English Wiktionary (Wikimedia Foundation, 2012). We mine entries for English words that explicitly have foreign translations listed as well as words in each target language that have English definitions. We discard all translation entries where the English side is longer than one word, except for constructions of the form "to VERB", where we manually remove the "to" and allow the word to be defined as the English infinitive. Finally, because our method requires a dictionary with probability weights, we assume that each target language word translates with uniform probability into any of the candidates that we scrape.

3.3 BANKS

We first evaluate our model under the BANKS data condition. Following the procedure from McDonald et al. (2011), for each language, we train both our DELEX and DELEX+PROJ features on a concatenation of 2000 sentences from each other CoNLL training set, plus 2000 sentences from the Penn Treebank. Again, despite the values of our PROJ queries being sensitive to which language we are currently parsing, the signatures are language independent, so discriminative training still makes sense over such a combined treebank. Training our PROJ features on the non-English treebanks in this concatenation can be understood as trying to learn which lexico-syntactic properties transfer "universally," or at least transfer broadly within the families of languages we are considering.

Table 1 shows the performance of the DELEX feature set and the DELEX+PROJ feature set using both AUTOMATIC and MANUAL bilingual lexicons. Both methods provide positive gains across the board that are statistically significant in the vast majority of cases, though MANUAL is slightly less effective; we postpone until Section 4.1 the discussion of the shortcomings of the MANUAL lexicon.

We include for reference the baseline results of McDonald et al. (2011) and Täckström et al. (2012) (multi-direct transfer and no clusters) and the improvements from their best methods using lexical information (multi-projected transfer and cross-lingual clusters). We emphasize that these results are *not* directly comparable to our own, as we have different training data (and even different training languages) and use a different underlying parsing model (MSTParser instead of a transition-based

AUTOMATIC									
	100 train trees			200 train trees			400 train trees		
	DELEX	DELEX+PROJ	Δ	DELEX	DELEX+PROJ	Δ	DELEX	DELEX+PROJ	Δ
DA	67.2	69.5	2.32 ‡	69.5	72.3	2.77 ‡	71.4	74.6	3.16 ‡
DE	72.9	73.9	0.97	75.4	76.5	1.09 †	77.3	78.5	1.25 ‡
EL	70.8	72.9	2.07 ‡	72.6	74.9	2.30 ‡	74.3	76.7	2.41 ‡
ES	72.5	73.0	0.46	74.1	75.4	1.29 ‡	75.3	77.2	1.81 ‡
IT	73.3	75.4	2.13 ‡	74.7	77.3	2.54 ‡	76.0	78.7	2.74 ‡
NL	63.0	65.8	2.82 ‡	64.7	67.6	2.86 ‡	66.1	69.2	3.06 ‡
PT	78.1	79.5	1.45 ‡	79.5	81.1	1.66 ‡	80.7	82.4	1.63 ‡
SV	76.4	78.1	1.69 ‡	78.1	80.2	2.02 ‡	79.6	81.7	2.07 ‡
AVG	71.8	73.5	1.74	73.6	75.7	2.07	75.1	77.4	2.27
EN	74.4	81.5	7.06 ‡	76.6	83.0	6.35 ‡	78.3	84.1	5.80 ‡
MANUAL									
DA	67.2	68.1	0.88	69.5	70.9	1.44 ‡	71.4	73.3	1.92 ‡
DE	72.9	73.4	0.44	75.4	76.2	0.77	77.3	78.4	1.12 ‡
EL	70.8	71.9	1.06 †	72.6	74.1	1.48 ‡	74.3	75.8	1.56 ‡
ES	72.5	71.9	-0.64	74.1	74.3	0.23	75.3	76.4	1.04 ‡
IT	73.3	74.3	1.01 †	74.7	76.4	1.66 ‡	76.0	78.0	2.01 ‡
NL	63.0	65.4	2.43 ‡	64.7	67.5	2.76 ‡	66.1	69.0	2.91 ‡
PT	78.1	78.2	0.13	79.5	80.1	0.62	80.7	81.5	0.82 ‡
SV	76.4	76.6	0.25	78.1	79.1	1.01 †	79.6	81.0	1.40 ‡
AVG	71.8	72.5	0.70	73.6	74.8	1.25	75.1	76.7	1.60
EN	74.4	81.5	7.06 ‡	76.6	83.0	6.35 ‡	78.3	84.1	5.80 ‡

Table 2: Evaluation of features derived from AUTOMATIC and MANUAL bilingual lexicons when trained on various small numbers of target language trees (the SEED setting). Values reported are UAS for sentences of all lengths on our enlarged CoNLL test sets (see text); each value is based on 50 sampled training sets of the given size. Daggers indicate statistical significance as described in the text. Statistical significance is not reported for averages.

parser (Nivre, 2008)). However, our baseline is competitive with theirs,[10] demonstrating that we have constructed a state-of-the-art delexicalized parser. Furthermore, our method appears to approach the performance of previous bitext-based methods, and because of its flexibility and the freedom from complex cross-lingual training for each new language, it can be applied in the MANUAL case as well, a capability which neither of the other methods has.

3.4 SEED

We now turn our attention to the SEED scenario, where a small number of target language trees are available for each language we consider. While it is imaginable to continue to exploit the other treebanks in the presence of target language trees, we found that training our DELEX features on the seed treebank alone gave higher performance than any attempt to also use the concatenation of treebanks from the previous section. This is not too surprising because, with this number of sentences, there is already good monolingual coverage of coarse POS features, and attempting to train features on other languages can be expected to introduce noise into otherwise accurate monolingual feature weights.

We train our DELEX+PROJ model with both AUTOMATIC and MANUAL lexicons on target language training sets of size 100, 200, and 400, and give results for each language in Table 2. The performance of parsers trained on small numbers of trees can be highly variable, so we create multiple treebanks of each size by repeatedly sampling from each language's train treebank, and report averaged results. Furthermore, this evaluation is not on the standard CoNLL test sets, but is instead on those test sets with a few hundred unused training sentences added, the reason being that some of the CoNLL test sets are very small (fewer than 200 sentences) and appeared

[10] The baseline of Täckström et al. (2012) is lower because it is trained only on English rather than on many languages.

to give highly variable results. To compute statistical significance, we draw a large number of bootstrap samples for each training set used, then aggregate all of their sufficient statistics in order to compute the final p-value. We see that our DELEX+PROJ method gives statistically significant gains at the 95% level over DELEX for nearly all language and training set size pairs, giving on average a 9% relative error reduction in the 400-tree case.

Because our features are relatively few in number and capture heuristic information, one question we might ask is how well they can perform in a non-projection context. In the last line of the table, we report gains that are achieved when PROJ features computed from parsed Gigaword are used directly on English, with no intermediate dictionary. These are not comparable to the other values in the table because we are using our projection strategy monolingually, which removes the barriers of imperfect lexical correspondence (from using the lexicon) and imperfect syntactic correspondence (from projecting). As one might expect, the gains on English are far higher than the gains on other languages. This indicates that performance is chiefly limited by the need to do cross-lingual feature adaptation, not inherently low feature capacity. We delay further discussion to Section 4.2.

One surprising thing to note is that the gains given by our PROJ features are in some cases larger here than in the BANKS setting. This result is slightly counterintuitive, as our baseline parsers are much better in this case and so we would expect diminished returns from our method. We conclude that accurately learning which signatures transfer between languages is important, and it is easier to learn good feature weights when some target language data is available. Further evidence supporting this hypothesis is the fact that the gains are larger and more significant on larger training set sizes.

4 Discussion

4.1 AUTOMATIC versus MANUAL

Overall, we see that gains from using our MANUAL lexicons are slightly lower than those from our AUTOMATIC lexicons. One might expect higher performance because scraped bilingual lexicons are not prone to some of the same noise that exists in automatic aligners, but this is empirically not the case. Rather, as we see in Table 3, the low recall of our MANUAL lexicons on open-class words appears to be a possible culprit. The coverage gap between these and the AUTOMATIC lexicons is partially due to the inconsistent structure of Wiktionary: inflected German and Greek words often do not have their own pages, so we miss even common morphological variants of verb forms in those languages. The inflected forms that we do scrape are also mapped to the English base form rather than the corresponding inflected form in English, which introduces further noise. Coverage is substantially higher if we translate using stems only, but this did not empirically lead to performance improvements, possibly due to conflating different parts of speech with the same base form.

	AUTOMATIC		MANUAL	
	Voc	OCC	Voc	OCC
DA	324K	0.91	22K	0.64
DE	320K	0.89	58K	0.55
EL	196K	0.94	23K	0.43
ES	165K	0.89	206K	0.74
IT	158K	0.91	78K	0.65
NL	251K	0.87	50K	0.72
PT	165K	0.85	46K	0.53
SV	307K	0.93	28K	0.60

Table 3: Lexicon statistics for all languages for both sources of bilingual lexicons. "Voc" indicates vocabulary size and "OCC" indicates open-class coverage, the fraction of open-class tokens in the test treebanks with entries in our bilingual lexicon.

One might hypothesize that our uniform weighting scheme in the MANUAL lexicon is another source of problems, and that bitext-derived weights are necessary to get high performance. This is not the case here. Truncating the AUTOMATIC dictionary to at most 20 translations per word and setting the weights uniformly causes a slight performance drop, but is still better than our MANUAL lexicon. This further demonstrates that these problems are more a limitation of our dictionary than our method. English Wiktionary is not designed to be a bilingual dictionary, and while it conveniently provided an easy way for us to produce lexicons for a wide array

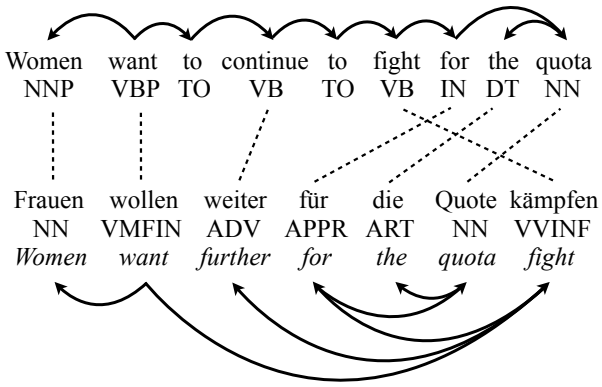

Figure 4: Example of a German tree and a parallel English sentence with high levels of syntactic divergence. The English verb *want* takes fundamentally different children than *wollen* does, so properties of the sort we present in Section 2.2 will not transfer effectively.

of languages, it is not the resource that one would choose if designing a parser for a specific target language. Bitext is not necessary for our approach to work, and results on the AUTOMATIC lexicon suggest that our type-level transfer method can in fact do much better given a higher quality resource.

4.2 Limitations

While our method does provide consistent gains across a range of languages, the injection of lexical information is clearly not sufficient to bridge the gap between unsupervised and supervised parsers. We argued in Section 3.4 that the cross-lingual transfer step of our method imposes a fundamental limitation on how useful any such approach can be, which we now investigate further.

In particular, any syntactic divergence, especially inconsistent divergences like head switching, will limit the utility of transferred structure. Consider the German example in Figure 4, with a parallel English sentence provided. The English tree suggests that *want* should attach to an infinitival *to*, which has no correlate in German. Even disregarding this, its grandchild is the verb *continue*, which is realized in the German sentence as the adverb *weiter*. While it is still broadly true that *want* and *wollen* both have verbal elements located to their right, it is less clear how to design features that can still take advantage of this while working around the differences we have described. Therefore, a gap between the performance of our features on English and the performance of our projected features, as is observed in Table 2, is to be expected in the absence of a more complete model of syntactic divergence.

5 Conclusion

In this work, we showed that lexical attachment preferences can be projected to a target language at the type level using only a bilingual lexicon, improving over a delexicalized baseline parser. This method is broadly applicable in the presence or absence of target language training trees and with bilingual lexicons derived from either manually-annotated resources or bitexts. The greatest improvements arise when the bilingual lexicon has high coverage and a number of target language trees are available in order to learn exactly what lexico-syntactic properties transfer from the source language.

In addition, we showed that a well-tuned discriminative model with the correct features can achieve good performance even on very small training sets. While unsupervised and existing projection methods do feature great versatility and may yet produce state-of-the-art parsers on resource-poor languages, spending time constructing small supervised resources appears to be the fastest method to achieve high performance in these settings.

Acknowledgments

This work was partially supported by an NSF Graduate Research Fellowship to the first author, by a Google Fellowship to the second author, and by the NSF under grant 0643742. Thanks to the anonymous reviewers for their insightful comments.

References

Mohit Bansal and Dan Klein. 2011. Web-scale Features for Full-scale Parsing. In *Proceedings of ACL*, pages 693–702, Portland, Oregon, USA.

Taylor Berg-Kirkpatrick and Dan Klein. 2010. Phylogenetic Grammar Induction. In *Proceedings of ACL*, pages 1288–1297, Uppsala, Sweden.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of CoNLL*, pages 149–164.

Shay B. Cohen and Noah A. Smith. 2009. Shared Logistic Normal Distributions for Soft Parameter Tying in

Unsupervised Grammar Induction. In *Proceedings of NAACL*, pages 74–82, Boulder, Colorado.

Shay B. Cohen, Dipanjan Das, and Noah A. Smith. 2011. Unsupervised Structure Prediction with Non-Parallel Multilingual Guidance. In *Proceedings of EMNLP*, pages 50–61, Edinburgh, UK.

Michael Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. *Ph.D. thesis, University of Pennsylvania*.

Koby Crammer and Yoram Singer. 2001. Ultraconservative Online Algorithms for Multiclass Problems. *Journal of Machine Learning Research*, 3:2003.

Dipanjan Das and Slav Petrov. 2011. Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections. In *Proceedings of ACL*, pages 600–609, Portland, Oregon, USA.

Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency Grammar Induction via Bitext Projection Constraints. In *Proceedings of ACL*, pages 369–377, Suntec, Singapore.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2007. English Gigaword Third Edition. Linguistic Data Consortium, Catalog Number LDC2007T07.

Aria Haghighi and Dan Klein. 2006. Prototype-driven Grammar Induction. In *Proceedings of CoLING-ACL*, pages 881–888, Sydney, Australia.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping Parsers via Syntactic Projection Across Parallel Texts. *Natural Language Engineering*, 11:311–325, September.

Dan Klein and Christopher D. Manning. 2004. Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency. In *Proceedings of ACL*, pages 479–486.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit X*, pages 79–86, Phuket, Thailand. AAMT.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple Semi-Supervised Dependency Parsing. In *Proceedings of ACL*.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by Agreement. In *Proceedings of NAACL*, New York, New York.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330, June.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online Large-Margin Training of Dependency Parsers. In *Proceedings of ACL*, pages 91–98, Ann Arbor, Michigan.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-Source Transfer of Delexicalized Dependency Parsers. In *Proceedings of EMNLP*, pages 62–72, Edinburgh, Scotland, UK.

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using Universal Linguistic Knowledge to Guide Grammar Induction. In *Proceedings of EMNLP*, pages 1234–1244, Cambridge, Massachusetts.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan Mcdonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of EMNLP-CoNLL*, pages 915–932, Prague, Czech Republic.

Joakim Nivre. 2008. Algorithms for Deterministic Incremental Dependency Parsing. *Computational Linguistics*, 34:513–553, December.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of ACL*, pages 433–440, Sydney, Australia.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A Universal Part-of-Speech Tagset. In *ArXiv*, April.

David A. Smith and Jason Eisner. 2009. Parser Adaptation and Projection with Quasi-Synchronous Grammar Features. In *Proceedings of EMNLP*, pages 822–831, Suntec, Singapore.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual Word Clusters for Direct Transfer of Linguistic Structure. In *Proceedings of NAACL*, Montreal, Canada.

Wikimedia Foundation. 2012. Wiktionary. Online at http://www.wiktionary.org/.

Guangyou Zhou, Jun Zhao, Kang Liu, and Li Cai. 2011. Exploiting Web-Derived Selectional Preference to Improve Statistical Dependency Parsing. In *Proceedings of ACL*, pages 1556–1565, Portland, Oregon, USA.