Rich bitext projection features for parse reranking

Alexander Fraser **Renjing Wang** **Hinrich Schütze**
Institute for Natural Language Processing
University of Stuttgart
{fraser,wangrg}@ims.uni-stuttgart.de

Abstract

Many different types of features have been shown to improve accuracy in parse reranking. A class of features that thus far has not been considered is based on a projection of the syntactic structure of a translation of the text to be parsed. The intuition for using this type of *bitext projection feature* is that ambiguous structures in one language often correspond to unambiguous structures in another. We show that reranking based on bitext projection features increases parsing accuracy significantly.

1 Introduction

Parallel text or *bitext* is an important knowledge source for solving many problems such as machine translation, cross-language information retrieval, and the projection of linguistic resources from one language to another. In this paper, we show that bitext-based features are effective in addressing another NLP problem, increasing the accuracy of statistical parsing. We pursue this approach for a number of reasons. First, one limiting factor for syntactic approaches to statistical machine translation is parse quality (Quirk and Corston-Oliver, 2006). Improved parses of bitext should result in improved machine translation. Second, as more and more texts are available in several languages, it will be increasingly the case that a text to be parsed is itself part of a bitext. Third, we hope that the improved parses of bitext will serve as higher quality training data for improving monolingual parsing using a process similar to self-training (McClosky et al., 2006).

It is well known that different languages encode different types of grammatical information (agreement, case, tense etc.) and that what can be left unspecified in one language must be made explicit in another. This information can be used for syntactic disambiguation. However, it is surprisingly hard to do this well. We use parses and alignments that are automatically generated and hence imperfect. German parse quality is considered to be worse than English parse quality, and the annotation style is different, e.g., NP structure in German is flatter.

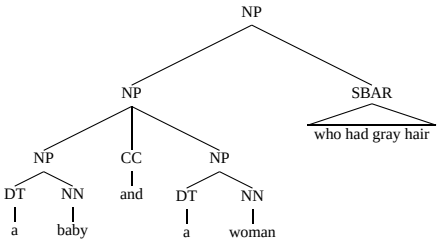

Figure 1: English parse with high attachment

We conduct our research in the framework of N-best parse reranking, but apply it to bitext and add only features based on *syntactic projection* from German to English. We test the idea that, generally, English parses with more isomorphism with respect to the projected German parse are better. The system takes as input (i) English sentences with a list of automatically generated syntactic parses, (ii) a translation of the English sentences into German, (iii) an automatically generated parse of the German translation, and (iv) an automatically generated word alignment. We achieve a significant improvement of 0.66 F_1 (absolute) on test data.

The paper is organized as follows. Section 2 outlines our approach and section 3 introduces the model. Section 4 describes training and section 5 presents the data and experimental results. In section 6, we discuss previous work. Section 7 analyzes our results and section 8 concludes.

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 282–290,
Athens, Greece, 30 March – 3 April 2009.

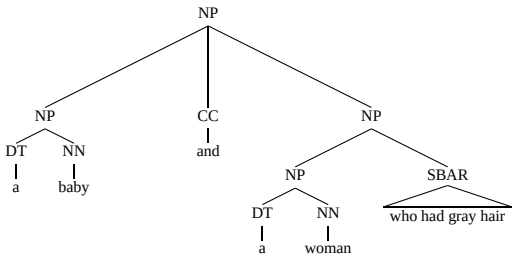

Figure 2: English parse with low attachment

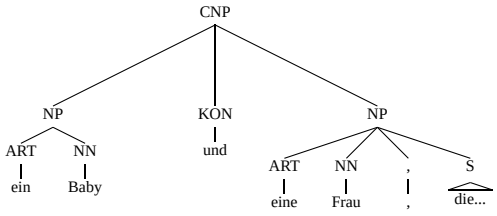

Figure 3: German parse with low attachment

2 Approach

Consider the English sentence "He saw a baby and a woman who had gray hair". Suppose that the baseline parser generates two parses, containing the NPs shown in figures 1 and 2, respectively, and that the semantically more plausible second parse in figure 2 is correct. How can we determine that the second parse should be favored? Since we are parsing bitext, we can observe the German translation which is "Er sah ein Baby und eine Frau, die graue Haare hatte" (glossed: "he saw a baby and a woman, who gray hair had"). The singular verb in the subordinate clause ("hatte": "had") indicates that the subordinate S must be attached low to "woman" ("Frau") as shown in figure 3.

We follow Collins' (2000) approach to discriminative reranking (see also (Riezler et al., 2002)). Given a new sentence to parse, we first select the best N parse trees according to a generative model. Then we use new features to learn discriminatively how to rerank the parses in this N-best list. We use features derived using projections of the 1-best German parse onto the hypothesized English parse under consideration.

In more detail, we take the 100 best English parses from the BitPar parser (Schmid, 2004) and rerank them. We have a good chance of finding the optimal parse among the 100-best[1]. An automatically generated word alignment determines translational correspondence between German and English. We use features which measure *syntactic divergence* between the German and English trees to try to rank the English trees which have less divergence higher. Our test set is 3718 sentences from the English Penn treebank (Marcus et al., 1993) which were translated into German. We hold out these sentences, and train BitPar on the remaining Penn treebank training sentences. The average F_1 parsing accuracy of BitPar on this test set is 87.89%, which is our baseline[2]. We implement features based on projecting the German parse to each of the English 100-best parses in turn via the word alignment. By performing cross-validation and measuring test performance within each fold, we compare our new system with the baseline on the 3718 sentence set. The overall test accuracy we reach is 88.55%, a statistically significant improvement over baseline of 0.66.

Given a word alignment of the bitext, the system performs the following steps for each English sentence to be parsed:
(i) run BitPar trained on English to generate 100-best parses for the English sentence
(ii) run BitPar trained on German to generate the 1-best parse for the German sentence
(iii) calculate feature function values which measure different kinds of syntactic divergence
(iv) apply a model that combines the feature function values to score each of the 100-best parses
(v) pick the best parse according to the model

3 Model

We use a log-linear model to choose the best English parse. The feature functions are functions on the hypothesized English parse e, the German parse g, and the word alignment a, and they assign a score (varying between 0 and infinity) that measures *syntactic divergence*. The alignment of a sentence pair is a function that, for each English word, returns a set of German words that the English word is aligned with as shown here for the sentence pair from section 2:
Er sah ein Baby und eine Frau , die graue Haare hatte
He{1} saw{2} a{3} baby{4} and{5} a{6} woman{7} who{9} had{12} gray{10} hair{11}

Feature function values are calculated either by taking the negative $\log$ of a probability, or by using a heuristic function which scales in a similar fash-

[1]Using an oracle to select the best parse results in an F_1 of 95.90, an improvement of 8.01 absolute over the baseline.

[2]The test set is very challenging, containing English sentences of up to 99 tokens.

ion[3]. The form of the log-linear model is shown in eq. 1. There are M feature functions $h_1, \ldots, h_M$. The vector λ is used to control the contribution of each feature function.

$$p_\lambda(e|g,a) = \frac{exp(-\sum_i \lambda_i h_i(e,g,a))}{\sum_{e'} exp(-\sum_i \lambda_i h_i(e',g,a))} \quad (1)$$

Given a vector of weights λ, the best English parse $\hat{e}$ can be found by solving eq. 2. The model is trained by finding the weight vector λ which maximizes accuracy (see section 4).

$$\begin{aligned} \hat{e} &= \operatorname*{argmax}_e p_\lambda(e|g,a) \\ &= \operatorname*{argmin}_e exp(\sum_i \lambda_i h_i(e,g,a)) \end{aligned} \quad (2)$$

3.1 Feature Functions

The basic idea behind our feature functions is that any constituent in a sentence should play approximately the same syntactic role and have a similar span as the corresponding constituent in a translation. If there is an obvious disagreement, it is probably caused by wrong attachment or other syntactic mistakes in parsing. Sometimes in translation the syntactic role of a given semantic constitutent changes; we assume that our model penalizes all hypothesized parses equally in this case. For the initial experiments, we used a set of 34 probabilistic and heuristic feature functions.

BitParLogProb (the only monolingual feature) is the negative log probability assigned by BitPar to the English parse. If we set $\lambda_1 = 1$ and $\lambda_i = 0$ for all $i \neq 1$ and evaluate eq. 2, we will select the parse ranked best by BitPar.

In order to define our feature functions, we first introduce auxiliary functions operating on individual word positions or sets of word positions. Alignment functions take an alignment a as an argument. In the descriptions of these functions we omit a as it is held constant for a sentence pair (i.e., an English sentence and its German translation).

$f(i)$ returns the set of word positions of German words aligned with an English word at position i.

$f'(i)$ returns the leftmost word position of the German words aligned with an English word at position i, or zero if the English word is unaligned.

$f^{-1}(i)$ returns the set of positions of English words aligned with a German word at position i.

$f'^{-1}(i)$ returns the leftmost word position of the English words aligned with a German word at position i, or zero if the German word is unaligned.

We overload the above functions to allow the argument i to be a set, in which case union is used, for example, $f(i) = \cup_{j \in i} f(j)$. Positions in a tree are denoted with integers. First, the POS tags are numbered from 1 to the length of the sentence (i.e., the same as the word positions). Constituents higher in the tree are also indexed using consecutive integers. We refer to the constituent that has been assigned index i in the tree t as "constituent i in tree t" or simply as "constituent i". The following functions have the English and German trees as an implicit argument; it should be obvious from the argument to the function whether the index i refers to the German tree or the English tree. When we say "constituents", we include nodes on the POS level of the tree. Our syntactic trees are annotated with a syntactic head for each constituent. Finally, the tag at position 0 is NULL.

mid2sib(i) returns 0 if i is 0, returns 1 if i has exactly two siblings, one on the left of i and one on the right, and otherwise returns 0.

head(i) returns the index of the head of i. The head of a POS tag is its own position.

tag(i) returns the tag of i.

left(i) returns the index of the leftmost sibling of i.

right(i) returns the index of the rightmost sibling.

up(i) returns the index of i's parent.

$\Delta(i)$ returns the set of word positions covered by i. If i is a set, Δ returns all word positions between the leftmost position covered by any constituent in the set and the rightmost position covered by any constituent in the set (inclusive).

$n(A)$ returns the size of the set A.

$c(A)$ returns the number of characters (including punctuation and excluding spaces) covered by the constituents in set A.

$[\![\pi]\!]$ is 1 if π is true, and 0 otherwise.

l and m are the lengths in words of the English and German sentences, respectively.

3.1.1 Count Feature Functions

Feature **CrdBin** counts binary events involving the heads of coordinated phrases. If in the English parse we have a coordination where the English CC is aligned only with a German KON, and both have two siblings, then the value contributed to **CrdBin** is 1 (indicating a constraint violation) un-

[3] For example, a probability of 1 is a feature value of 0, while a low probability is a feature value which is $\gg 0$.

less the head of the English left conjunct is aligned with the head of the German left conjunct and likewise the right conjuncts are aligned. Eq. 3 calculates the value of **CrdBin**.

$$\sum_{i=1}^{l} [\![(\text{tag}(i) = \text{CC}]\!][\![(n(f(i)) = 1]\!] \quad \text{mid2sib}(i)$$
$$\text{mid2sib}(f'(i)) \quad [\![\text{tag}(f'(i)) = \text{KON-CD}]\!]$$
$$[\![[\text{head}(\text{left}(f'(i))) \neq f'(\text{head}(\text{left}(i)))] \text{ OR}$$
$$[\text{head}(\text{right}(f'(i))) \neq f'(\text{head}(\text{right}(i)))]]\!] \quad (3)$$

Feature **Q** simply captures a mismatch between questions and statements. If an English sentence is parsed as a question but the parallel German sentence is not, or vice versa, the feature value is 1; otherwise the value is 0.

3.1.2 Span Projection Feature Functions

Span projection features calculate the percentage difference between a constituent's span and the span of its projection. Span size is measured in characters or words. To project a constituent in a parse, we use the word alignment to project all word positions covered by the constituent and then look for the smallest covering constituent in the parse of the parallel sentence.

CrdPrj is a feature that measures the divergence in the size of coordination constituents and their projections. If we have a constituent (XP1 CC XP2) in English that is projected to a German coordination, we expect the English and German left conjuncts to span a similar percentage of their respective sentences, as should the right conjuncts. The feature computes a character-based percentage difference as shown in eq. 4.

$$\sum_{i=1}^{l} [\![\text{tag}(i) = \text{CC}]\!][\![n(f(i)) = 1]\!] \quad (4)$$
$$[\![\text{tag}(f'(i)) = \text{KON-CD}]\!]$$
$$\text{mid2sib}(i)\text{mid2sib}(f'(i))$$
$$(|\frac{c(\Delta(\text{left}(i)))}{r} - \frac{c(\Delta(\text{left}(f'(i))))}{s}|$$
$$+|\frac{c(\Delta(\text{right}(i)))}{r} - \frac{c(\Delta(\text{right}(f'(i))))}{s}|)$$

r and s are the lengths in characters of the English and German sentences, respectively. In the English parse in figure 1, the left conjunct has 5 characters and the right conjunct has 6, while in figure 2 the left conjunct has 5 characters and the right conjunct has 20. In the German parse (figure 3) the left conjunct has 7 characters and the right conjunct has 27. Finally, $r = 33$ and $s = 42$. Thus, the value of **CrdPrj** is 0.48 for the first hypothesized parse and 0.05 for the second, which captures the higher divergence of the first English parse from the German parse.

POSParentPrj is based on computing the span difference between all the parent constituents of POS tags in a German parse and their respective coverage in the corresponding hypothesized parse. The feature value is the sum of all the differences. $\text{POSPar}(i)$ is true if i immediately dominates a POS tag. The projection direction is from German to English, and the feature computes a percentage difference which is character-based. The value of the feature is calculated in eq. 5, where M is the number of constituents (including POS tags) in the German tree.

$$\sum_{i=1}^{M} [\![\text{POSPar}(i)]\!]|\frac{c(\Delta(i))}{s} - \frac{c(\Delta(f^{-1}(\Delta(i))))}{r}| \quad (5)$$

The right conjunct in figure 3 is a POSParent that corresponds to the coordination NP in figure 1, contributing a score of 0.21, and to the right conjunct in figure 2, contributing a score of 0.04. For the two parses of the full sentences containing the NPs in figure 1 and figure 2, we sum over 7 POSParents and get a value of 0.27 for parse 1 and 0.11 for parse 2. The lower value for parse 2 correctly captures the fact that the first English parse has higher divergence than the second due to incorrect high attachment.

AbovePOSPrj is similar to **POSParentPrj**, but it is word-based and the projection direction is from English to German. Unlike **POSParentPrj** the feature value is calculated over all constituents above the POS level in the English tree.

Another span projection feature function is **DTNNPrj**, which projects English constituents of the form (NP(DT)(NN)). $\text{DTNN}(i)$ is true if i is an NP immediately dominating only DT and NN. The feature computes a percentage difference which is word-based, shown in eq. 6.

$$\sum_{i=1}^{L} [\![\text{DTNN}(i)]\!]|\frac{n(\Delta(i))}{l} - \frac{n(\Delta(f(\Delta(i))))}{m}| \quad (6)$$

L is the number of constituents in the English tree. This feature is designed to disprefer parses

where constituents starting with "DT NN", e.g., (NP (DT NN NN NN)), are incorrectly split into two NPs, e.g., (NP (DT NN)) and (NP (NN NN)). This feature fires in this case, and projects the (NP (DT NN)) into German. If the German projection is a surprisingly large number of words (as should be the case if the German also consists of a determiner followed by several nouns) then the penalty paid by this feature is large. This feature is important as (NP (DT NN)) is a very common construction.

3.1.3 Probabilistic Feature Functions

We use Europarl (Koehn, 2005), from which we extract a parallel corpus of approximately 1.22 million sentence pairs, to estimate the probabilistic feature functions described in this section.

For the **PDepth** feature, we estimate English parse depth probability conditioned on German parse depth from Europarl by calculating a simple probability distribution over the 1-best parse pairs for each parallel sentence. A very deep German parse is unlikely to correspond to a flat English parse and we can penalize such a parse using **PDepth**. The index i refers to a sentence pair in Europarl, as does j. Let l_i and m_i be the depths of the top BitPar ranked parses of the English and German sentences, respectively. We calculate the probability of observing an English tree of depth l' given German tree of depth m' as the maximum likelihood estimate, shown in eq. 7, where $\delta(z, z') = 1$ if $z = z'$ and 0 otherwise. To avoid noisy feature values due to outliers and parse errors, we bound the value of **PDepth** at 5 as shown in eq. 8[4].

$$p(l'|m') = \frac{\sum_i \delta(l', l_i)\delta(m', m_i)}{\sum_j \delta(m', m_j)} \quad (7)$$

$$\min(5, -\log_{10}(p(l'|m'))) \quad (8)$$

The full parse of the sentence containing the English high attachment has a parse depth of 8 while the full parse of the sentence containing the English low attachment has a depth of 9. Their feature values given the German parse depth of 6 are $-\log_{10}(0.12) = 0.93$ and $-\log_{10}(0.14) = 0.84$. The wrong parse is assigned a higher feature value indicating its higher divergence.

The feature **PTagEParentGPOSGParent** measures tagging inconsistency based on estimating the probability that for an English word at position i, the parent of its POS tag has a particular label. The feature value is calculated in eq. 10.

$$q(i, j) = p(\text{tag}(\text{up}(i))|\text{tag}(j), \text{tag}(\text{up}(j))) \quad (9)$$

$$\sum_{i=1}^{l} \min(5, \frac{\sum_{j \in f(i)} -\log_{10}(q(i, j))}{n(f(i))}) \quad (10)$$

Consider (S(NP(NN fruit))(VP(V flies))) and (NP(NN fruit)(NNS flies)) with the translation (NP(NNS Fruchtfliegen)). Assume that "fruit" and "flies" are aligned with the German compound noun "Fruchtfliegen". In the incorrect English parse the parent of the POS of "fruit" is NP and the parent of the POS of "flies" is VP, while in the correct parse the parent of the POS of "fruit" is NP and the parent of the POS of "flies" is NP. In the German parse the compound noun is POS-tagged as an NNS and the parent is an NP. The probabilities considered for the two English parses are $p(\text{NP}|\text{NNS}, \text{NP})$ for "fruit" in both parses, $p(\text{VP}|\text{NNS}, \text{NP})$ for "flies" in the incorrect parse, and $p(\text{NP}|\text{NNS}, \text{NP})$ for "flies" in the correct parse. A German NNS in an NP has a higher probability of being aligned with a word in an English NP than with a word in an English VP, so the second parse will be preferred.

As with the **PDepth** feature, we use relative frequency to estimate this feature. When an English word is aligned with two words, estimation is more complex. We heuristically give each English and German pair one count. The value calculated by the feature function is the geometric mean[5] of the pairwise probabilities, see eq. 10.

3.1.4 Other Features

Our best system uses the nine features we have described in detail so far. In addition, we implemented the following 25 other features, which did not improve performance (see section 7): (i) 7 "ptag" features similar to **PTagEParentGPOSGParent** but predicting and conditioning on different combinations of tags (POS tag, parent of POS, grandparent of POS)

(ii) 10 "prj" features similar to **POSParentPrj** measuring different combinations of character and word percentage differences at the POS parent and

[4] Throughout this paper, assume $\log(0) = -\infty$.

[5] Each English word has the same weight regardless of whether it was aligned with one or with more German words.

POS grandparent levels, projecting from both English and German

(iii) 3 variants of the **DTNN** feature function

(iv) A **NPPP** feature function, similar to the **DTNN** feature function but trying to counteract a bias towards (NP (NP) (PP)) units

(v) A feature function which penalizes aligning clausal units to non-clausal units

(vi) The BitPar rank

4 Training

Log-linear models are often trained using the Maximum Entropy criterion, but we train our model directly to maximize F_1. We score F_1 by comparing hypothesized parses for the discriminative training set with the gold standard. To try to find the optimal λ vector, we perform direct accuracy maximization, meaning that we search for the λ vector which directly optimizes F_1 on the training set.

Och (2003) has described an efficient exact one-dimensional accuracy maximization technique for a similar search problem in machine translation. The technique involves calculating an explicit representation of the piecewise constant function $g_m(x)$ which evaluates the accuracy of the hypotheses which would be picked by eq. 2 from a set of hypotheses if we hold all weights constant, except for the weight λ_m, which is set to x. This is calculated in one pass over the data.

The algorithm for training is initialized with a choice for λ and is described in figure 4. The function $F_1(\lambda)$ returns F_1 of the parses selected using λ. Due to space we do not describe step 8 in detail (see (Och, 2003)). In step 9 the algorithm performs approximate normalization, where feature weights are forced towards zero. The implementation of step 9 is straight-forward given the M explicit functions $g_m(x)$ created in step 8.

1: **Algorithm** TRAIN(λ)
2: **repeat**
3: add λ to the set s
4: let t be a set of 1000 randomly generated vectors
5: let $\lambda = \text{argmax}_{\rho \in (s \cup t)} F_1(\rho)$
6: let $\lambda' = \lambda$
7: **repeat**
8: repeatedly run one-dimensional error minimization step (updating a single scalar of the vector λ) until no further error reduction
9: adjust each scalar of λ in turn towards 0 such that there is no increase in error (if possible)
10: **until** no scalar in λ changes in last two steps (8 and 9)
11: **until** $\lambda = \lambda'$
12: return λ

Figure 4: Sketch of the training algorithm

5 Data and Experiments

We used the subset of the Wall Street Journal investigated in (Atterer and Schütze, 2007) for our experiments, which consists of all sentences that have at least one prepositional phrase attachment ambiguity. This difficult subset of sentences seems particularly interesting when investigating the potential of information in bitext for improving parsing performance. The first 500 sentences of this set were translated from English to German by a graduate student and an additional 3218 sentences by a translation bureau. We withheld these 3718 English sentences (and an additional 1000 reserved sentences) when we trained BitPar on the Penn treebank.

Parses. We use the BitPar parser (Schmid, 2004) which is based on a bit-vector implementation (cf. (Graham et al., 1980)) of the Cocke-Younger-Kasami algorithm (Kasami, 1965; Younger, 1967). It computes a compact parse forest for all possible analyses. As all possible analyses are computed, any number of best parses can be extracted. In contrast, other treebank parsers use sophisticated search strategies to find the most probable analysis without examining the set of all possible analyses (Charniak et al., 1998; Klein and Manning, 2003). BitPar is particularly useful for N-best parsing as the N-best parses can be computed efficiently.

For the 3718 sentences in the translated set, we created 100-best English parses and 1-best German parses. The German parser was trained on the TIGER treebank. For the Europarl corpus, we created 1-best parses for both languages.

Word Alignment. We use a word alignment of the translated sentences from the Penn treebank, as well as a word alignment of the Europarl corpus. We align these two data sets together with data from the JRC Acquis (Steinberger et al., 2006) to try to obtain better quality alignments (it is well known that alignment quality improves as the amount of data increases (Fraser and Marcu, 2007)). We aligned approximately 3.08 million sentence pairs. We tried to obtain better alignment quality as alignment quality is a problem in many cases where syntactic projection would otherwise work well (Fossum and Knight, 2008).

	System	Train	+base	Test	+base
1	Baseline	87.89		87.89	
2	Contrastive (5 trials/fold)	88.70	0.82	88.45	0.56
3	Contrastive (greedy selection)	88.82	0.93	88.55	0.66

Table 1: Average F_1 of 7-way cross-validation

To generate the alignments, we used Model 4 (Brown et al., 1993), as implemented in GIZA++ (Och and Ney, 2003). As is standard practice, we trained Model 4 with English as the source language, and then trained Model 4 with German as the source language, resulting in two Viterbi alignments. These were combined using the *Grow Diag Final And* symmetrization heuristic (Koehn et al., 2003).

Experiments. We perform 7-way cross-validation on 3718 sentences. In each fold of the cross-validation, the training set is 3186 sentences, while the test set is 532 sentences. Our results are shown in table 1. In row 1, we take the hypothesis ranked best by BitPar. In row 2, we train using the algorithm outlined in section 4. To cancel out any effect caused by a particularly effective or ineffective starting λ value, we perform 5 trials each time. Columns 3 and 5 report the improvement over the baseline on train and test respectively. We reach an improvement of 0.56 over the baseline using the algorithm as described in section 4.

Our initial experiments used many highly correlated features. For our next experiment we use greedy feature selection. We start with a λ vector that is zero for all features, and then run the error minimization without the random generation of vectors (figure 4, line 4). This means that we add one feature at a time. This greedy algorithm winds up producing a vector with many zero weights. In row 3 of table 1, we used the greedy feature selection algorithm and trained using F_1, resulting in a performance of 0.66 over the baseline which is our best result. We performed a planned one-tailed paired t-test on the F_1 scores of the parses selected by the baseline and this system for the 3718 sentences (parses were taken from the test portion of each fold). We found that there is a significant difference with the baseline ($t(3717) = 6.42$, $p < .01$). We believe that using the full set of 34 features (many of which are very similar to one another) made the training problem harder without improving the fit to the training data, and that greedy feature selection helps with this (see also section 7).

6 Previous Work

As we mentioned in section 2, work on parse reranking is relevant, but a vital difference is that we use features based only on *syntactic projection* of the two languages in a bitext. For an overview of different types of features that have been used in parse reranking see Charniak and Johnson (2005). Like Collins (2000) we use cross-validation to train our model, but we have access to much less data (3718 sentences total, which is less than 1/10 of the data Collins used). We use rich feature functions which were designed by hand to specifically address problems in English parses which can be disambiguated using the German translation.

Syntactic projection has been used to bootstrap treebanks in resource poor languages. Some examples of projection of syntactic parses from English to a resource poor language for which no parser is available are the works of Yarowsky and Ngai (2001), Hwa et al. (2005) and Goyal and Chatterjee (2006). Our work differs from theirs in that we are performing a parse reranking task in English using knowledge gained from German parses, and parsing accuracy is generally thought to be worse in German than in English.

Hopkins and Kuhn (2006) conducted research with goals similar to ours. They showed how to build a powerful generative model which flexibly incorporates features from parallel text in four languages, but were not able to show an improvement in parsing performance. After the submission of our paper for review, two papers outlining relevant work were published. Burkett and Klein (2008) describe a system for simultaneously improving Chinese and English parses of a Chinese/English bitext. This work is complementary to ours. The system is trained using gold standard trees in both Chinese and English, in contrast with our system which only has access to gold standard trees in English. Their system uses a tree alignment which varies within training, but this does not appear to make a large difference in performance. They use coarsely defined features which are language independent. We use several features similar to their two best performing sets of features, but in contrast with their work, we also define features which are specifically aimed at English disambiguation problems that we have observed can be resolved

using German parses. They use an in-domain Chinese parser and out-of-domain English parser, while for us the English parser is in-domain and the German parser is out-of-domain, both of which make improving the English parse more difficult. Their Maximum Entropy training is more appropriate for their numerous coarse features, while we use Minimum Error Rate Training, which is much faster. Finally, we are projecting from a single German parse which is a more difficult problem. Fossum and Knight (2008) outline a system for using Chinese/English word alignments to determine ambiguous English PP-attachments. They first use an oracle to choose PP-attachment decisions which are ambiguous in the English side of a Chinese/English bitext, and then build a classifier which uses information from a word alignment to make PP-attachment decisions. No Chinese syntactic information is required. We use automatically generated German parses to improve English syntactic parsing, and have not been able to find a similar phenomenon for which only a word alignment would suffice.

7 Analysis

We looked at the weights assigned during the cross-validation performed to obtain our best result. The weights of many of the 34 features we defined were frequently set to zero. We sorted the features by the number of times the relevant λ scalar was zero (i.e., the number of folds of the cross-validation for which they were zero; the greedy feature selection is deterministic and so we do not run multiple trials). We then reran the same greedy feature selection algorithm as was used in table 1, row 3, but this time using only the top 9 feature values, which were the features which were active on 4 or more folds[6]. The result was an improvement on train of 0.84 and an improvement on test of 0.73. This test result may be slightly overfit, but the result supports the inference that these 9 feature functions are the most important. We chose these feature functions to be described in detail in section 3. We observed that the variants of the similar features **POSParentPrj** and **AbovePOSPrj** projected in opposite directions and measured character and word differences, respectively, and this complementarity seems to help.

[6]We saw that many features canceled one another out on different folds. For instance either the word-based or the character-based version of **DTNN** was active in each fold, but never at the same time as one another.

We also tried to see if our results depended strongly on the log-linear model and training algorithm, by using the SVM-Light ranker (Joachims, 2002). In order to make the experiment tractable, we limited ourselves to the 8-best parses (rather than 100-best). Our training algorithm and model was 0.74 better than the baseline on train and 0.47 better on test, while SVM-Light was 0.54 better than baseline on train and 0.49 better on test (using linear kernels). We believe that the results are not unduly influenced by the training algorithm.

8 Conclusion

We have shown that rich bitext projection features can improve parsing accuracy. This confirms the hypothesis that the divergence in what information different languages encode grammatically can be exploited for syntactic disambiguation. Improved parsing due to bitext projection features should be helpful in syntactic analysis of bitexts (by way of mutual syntactic disambiguation) and in computing syntactic analyses of texts that have translations in other languages available.

Acknowledgments

This work was supported in part by Deutsche Forschungsgemeinschaft Grant SFB 732. We would like to thank Helmut Schmid for support of BitPar and for his many helpful comments on our work. We would also like to thank the anonymous reviewers.

References

Michaela Atterer and Hinrich Schütze. 2007. Prepositional phrase attachment without oracles. *Computational Linguistics*, 33(4).

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2).

David Burkett and Dan Klein. 2008. Two languages are better than one (for syntactic parsing). In *EMNLP*.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *ACL*.

Eugene Charniak, Sharon Goldwater, and Mark Johnson. 1998. Edge-based best-first chart parsing. In *Proceedings of the Sixth Workshop on Very Large Corpora*.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *ICML*.

Victoria Fossum and Kevin Knight. 2008. Using bilingual Chinese-English word alignments to resolve PP-attachment ambiguity in English. In *AMTA*.

Alexander Fraser and Daniel Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3).

Shailly Goyal and Niladri Chatterjee. 2006. Parsing aligned parallel corpus by projecting syntactic relations from annotated source corpus. In *Proceedings of the COLING/ACL main conference poster sessions*.

Susan L. Graham, Michael A. Harrison, and Walter L. Ruzzo. 1980. An improved context-free recognizer. *ACM Transactions on Programming Languages and Systems*, 2(3).

Mark Hopkins and Jonas Kuhn. 2006. A framework for incorporating alignment information in parsing. In *Proceedings of the EACL 2006 Workshop on Cross-Language Knowledge Induction*.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11(3).

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD*.

Takao Kasami. 1965. An efficient recognition and syntax analysis algorithm for context-free languages. Technical Report AFCRL-65-7558, Air Force Cambridge Research Laboratory.

Dan Klein and Christopher Manning. 2003. A* parsing: fast exact viterbi parse selection. In *HLT-NAACL*.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL*.

Philipp Koehn. 2005. Europarl: a parallel corpus for statistical machine translation. In *MT Summit X*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2).

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *HLT-NAACL*.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Franz J. Och. 2003. Minimum error rate training in statistical machine translation. In *ACL*.

Chris Quirk and Simon Corston-Oliver. 2006. The impact of parse quality on syntactically-informed statistical machine translation. In *EMNLP*.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard S. Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a lexical-functional grammar and discriminative estimation techniques. In *ACL*.

Helmut Schmid. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *COLING*.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Daniel Varga. 2006. The JRC-Acquis: a multilingual aligned parallel corpus with 20+ languages. In *LREC*.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *NAACL*.

Daniel H. Younger. 1967. Recognition of context-free languages in time n^3. *Information and Control*, 10.