Invited paper

A Progress Report on Bitext Parsing

Alexander Fraser
Institute for Natural Language Processing, University of Stuttgart
E-mail: fraser@ims.uni-stuttgart.de

Renjing Wang
Institute for Natural Language Processing, University of Stuttgart
E-mail: wangrg@ims.uni-stuttgart.de

Hinrich Schütze
Institute for Natural Language Processing, University of Stuttgart
Azenbergstrasse 11, 70180 Stuttgart, Germany

Summary

Recent work has shown that a reranking approach can be used to improve the syntactic parsing of a sentence given a translation of that sentence, an automatically generated parse of that translation, and a word alignment between them. Such approaches rely on reducing syntactic divergence as measured using overlapping feature functions capturing different types of divergence. These feature functions are combined in a log-linear model which is trained to maximize parsing accuracy.
We conduct our research in the framework of N-best parse reranking. However, we apply reranking to bitext and add only features based on syntactic projection from German to English. 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 between the original sentences and the translations. The system is trained using the gold standard trees of 3718 sentences from the Penn English treebank that have been translated into German. We achieve an improvement in F1 on held out test data and this improvement is statistically significant.

Key words: syntactic parsing, multilinguality, treebanks, machine translation, annotation projection

Introduction

Recent work [1,2] has shown that a reranking approach can be used to improve the syntactic parsing of a sentence given a translation of that sentence, an automatically generated parse of that translation, and a word alignment between them. Such approaches rely on reducing syntactic divergence as measured using overlapping feature functions capturing different types of divergence. These feature functions are combined in a log-linear model which is trained to maximize parsing accuracy.

In this work we extend the approach of Fraser, Wang and Schuetze [1]. We view this work as part of a research program aimed at finding alternative sources of supervision for syntactic parsing which can augment small and expensive to create syntactic treebanks. We analyze the gains in parsing accuracy obtained by this approach and provide examples, with a special focus on features which are estimated using the baseline parser on 1.4 million parallel sentences from the Europarl corpus (for which we do not have gold standard parses). This is similar to the self-training approach of McClosky, Charniak and Johnson [5]. We then augment the approach with two new feature functions which capture difficult prepositional phrase attachment phenomena, resulting in a further gain in performance as measured through cross-validation on sentences taken from the Penn Treebank. Finally, we discuss applying the system to the Europarl corpus and discuss possible improvements.

We conduct our research in the framework of N-best parse reranking (following Collins [6], but see also Riezler et. al. [7]). However, we apply reranking to bitext and add only features based on syntactic projection from German to English. The system takes as input:

1. English sentences with a list of automatically generated syntactic parses
2. A translation of the English sentences into German
3. An automatically generated parse of the German translation
4. An automatically generated word alignment between the original sentences and the translations

The system is trained using the gold standard trees of 3718 sentences from the Penn English treebank that have been translated into German. We achieve an improvement in F1 on held out test data (measured by using cross-validation) and this improvement is statistically significant.

Bitext Parsing

As a motivating example consider the English sentence "He saw a baby and a woman who had gray hair". Suppose that the baseline parser generates two parses, one where it is attached high (to both of the NPs), and one where "who had gray hair" is attached only to the woman. Suppose further, that the second parse is the correct parse in this context. 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") because the subject is singular. In correctly resolving the attachment ambiguity, we are using the human translator's disambiguation of the English syntax (performed while translating to German).

To accomplish this automatically, we follow Collins' approach [6] to discriminative reranking. The approach begins with a generative model which models the joint generation of a sentence and its parse tree and then reranks the 100-best hypothesized parses. 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. Because our features are based on bilingual projection, they are complementary to the features used in previous parse reranking work.

In more detail, we take the 100 best English parses from the BitPar parser [8] and rerank them. We have a good chance of finding the optimal parse among the 100-best hypothesized parses. 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 which were translated into German. We hold out these sentences, and train BitPar on the remaining Penn treebank training sentences. The average F1 parsing accuracy of BitPar on this test set is 87.89%, which is our baseline. The test set is very challenging, containing English sentences of up to 99 tokens.

We implement features based on projecting the German parse to each of the English 100-best parses in turn via the word alignment. All parses and the word alignment are generated automatically.

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 88.59%, a statistically significant improvement over baseline of 0.70.

Given a word alignment of the bitext, the system performs the following steps for each English sentence to be parsed:

1. Run BitPar trained on English to generate 100-best parses for the English sentence
2. Run BitPar trained on German to generate the 1-best parse for the German sentence
3. Calculate feature function values which measure different kinds of syntactic divergence
4. Apply a model that combines the feature function values to score each of the 100-best parses
5. Pick the best parse according to the model

Related Work

The most directly related work is that of Burkett and Klein [2], which is work that was published after [1] was submitted for publication. Similarly to our previous work, they used feature functions defined on triples of (English parse tree, Chinese parse tree, alignment) which were combined in a log-linear model. To train this model they used a small parallel treebank which contains gold standard trees for parallel sentences in Chinese and English, while we only required access to gold standard trees for the English side of our training corpus in order to improve English parse quality. They defined similar features to the coarse features defined in our previous work and trained a system which improves first the Chinese parse and then the English parse and iterates. In addition they try experiments allowing the alignment to vary, but these experiments are inconclusive. Our additional features go beyond the coarse syntactic divergence features in their work to address specific problems we observed through error analysis, and to incorporate self-training features. Two other interesting works in this area are those of Fossum and Knight [3]; and of Huang, Jiang and Liu [4]. They improve English prepositional phrase attachment using features from a Chinese sentence. However, unlike our approach, they do not require a Chinese syntactic parse as the word order in Chinese is sufficient to unambiguously determine the correct attachment point of the prepositional phrase in the English sentence without using a Chinese syntactic parse.

Model

We define feature functions which measure syntactic divergence. We use a model combining feature functions in a linear fashion, a log-linear model, to choose the best English parse (see the first equation below). The feature functions **h** 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. 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 fashion (for example, a probability of 1 is a feature value of 0, while a low probability is a feature value which is a large magnitude positive number. Note also that we define the value of log 0 to be –infinity for the purposes of this work, though in practice we do not work with probabilities of 0).

Given a vector of weights λ, the best English parse $\hat{e}$ can be found by solving the second equation below. The model is trained by finding the weight vector λ which maximizes accuracy. This is done by reranking the output of the generative model for a set of sentences for which we have gold standard parses. Training is discussed in detail later in the paper.

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

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

Training

Log-linear models are often trained using the Maximum Entropy criterion, but we train our model directly to maximize F1. We score F1 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 F1 on the training set, using the algorithm of [10]. See [1] for further details.

Feature Functions

We first briefly describe the feature functions we found useful in our previous work, and then introduce two new feature functions which we defined after an error analysis. 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 constituent 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, but we do not have space to briefly describe all 34 features.

BitPar LogProb (the only monolingual feature) is the negative log probability assigned by BitPar to the English parse. This feature is important, as it encodes the monolingually derived knowledge which is inherent in BitPar's model. The rest of the feature functions are bilingual and encode additional sources of knowledge derived from the parse of the German translation.

Count Feature Functions

We now introduce feature functions which *count* projection constraint violations.

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) unless the head of the English left conjunct is aligned with the head of the German left conjunct and likewise the right conjuncts are aligned.

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.

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.

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. The projection direction is from German to English, and the feature computes a percentage difference which is character-based.

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)). The feature computes a percentage difference which is word-based. It 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.

Probabilistic Feature Functions

We use Europarl corpus of Koehn [9], 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 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. 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(NP|NNS, NP) for "fruit" in both parses, p(VP|NNS, NP) for "flies" in the incorrect parse, and p(NP|NNS, 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.

Note that when an English word is aligned with two words, estimation is more complex. We heuristically give each English and German pair in the alignment unit one count. The value calculated by the feature function also works differently. If an English word is aligned with multiple German words, we use the geometric mean of the pairwise probabilities (i.e., each English word has the same overall weight regardless of whether it was aligned with one or with more German words).

Other Features

Our best system uses the nine features we have described in detail so far. In addition, we implemented 25 other features, which did not appear to improve performance, as we showed using a feature analysis in our previous work, see [1] for further details.

New Feature Functions

After conducting an error analysis of our system, we noticed that it had systematic failures in PP attachment which occurred higher in the tree than our previous feature functions had addressed. We define two new feature functions below, where we make use of the node numbering we introduced in [1] (briefly, all nodes in the tree including POS tags are assigned a unique integer; by convention **i** refers to a node in the English tree, and **j** refers to a node in the German tree). Recall also that higher values indicate penalized behaviour (these values scale like negative log probabilities).

The first feature **PPinNPPP c**hecks whether a PP inside of a NP or PP in German attaches to the same (projected) constituent in English.

```
For each German node j
  if j is PP and parent(i) is NP or PP
    let j' be the nearest sibling to the left of j that is a NN, NP or PP
    if j' is defined
      let English node i = project(j)
      let English node i' = project(j')
      value += 1 if i' is not a sibling of i
                          or i' not the nearest sibling to the left of i that is a NN, NP or PP
```

EngPPinSVP checks whether a PP inside of a S or VP in English attaches to the same (projected) constituent in German (note in the feature definition the attachment in the German can be to the left or to the right).

```
For each English node i
  if i is PP and parent(i) is S or VP
    let i' be nearest sibling to the left of i that is a POS(V*) or VP
    if i' is defined
      let German node j = project(i)
      let German node j' = project(i')
      value += 1 if j' is not POS(V*) or VP, or j' is not a sibling of j
```

Experiments

We used the subset of the Wall Street Journal which consists of all sentences that have at least one prepositional phrase attachment ambiguity for our experiments. An example of such an ambiguity is (VP bring (NP attention) (PP to the problem)) vs. (VP bring ((NP attention) (PP to the problem))). 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 [8] which is based on a bit-vector implementation of the Cocke-Younger-Kasami (CKY) algorithm. It computes a compact parse forest for all possible analyses. 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 to try to obtain better quality alignments. To generate word alignments, we used IBM Model 4 [13], as implemented in GIZA++ [11]. 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 [12].

Experimental Analysis

In [1] we reached the best performance by performing a greedy feature selection. We started with a λ vector that is zero for all features, and then ran the error minimization (without random generation of λ vectors, which makes the algorithm deterministic). One feature at a time is added. This greedy algorithm produced a vector with many zero weights, with a good fit to the training set resulting in a 0.93 improvement on the training set. The resulting performance of 0.66 on the test set over the baseline was our best result.

When we repeated this process using all of the features used in our previous work together with our two new features, we obtained an improvement of only 0.80 on the training set. On the test set, the improvement was only 0.55 over the baseline. This shows that with the addition of the 2 new features we are having problems with *search errors* for the λ vector which optimizes F1. We know that a vector exists which results in a better fit to the training data, it is simply the vector we had before, with the addition of two zeros added for the weights of our two new feature functions. Presumably, there might be another vector which assigns non-zero weights to our two new feature functions which will result in a further improvement.

We therefore went back to performing 5 trials per fold of our 7-fold cross-validation using the non-deterministic algorithm (these are combined by averaging). This algorithm differs in that it tries one thousand randomly determined λ vectors at the beginning of each iteration in an attempt to escape local minima which cannot be escaped from using the one-dimensional search. Using this algorithm our previous result was an improvement of 0.82 on train, and 0.55 on test. With the two new features we obtained an improved fit on train of 0.93 and an improvement on test of 0.70. This shows that the new features are effective. However, due to the search errors with the greedy algorithm we are unable to effectively apply it, even though we found it superior previously.

Conclusion

In this work we have introduced two new feature functions which improve over the system we presented in [1]. Although we had worse performance with the greedy feature selection algorithm we previously, performance with the non-deterministic algorithm improved by an additional 0.15% F1 resulting in a total improvement of 0.70% F1 over the baseline. We are currently preparing to apply this system to generate improved parses of the entire Europarl corpus, and we hope to obtain additional increases in parse quality through self-training [5] by retraining our baseline parser on the improved parses.

Acknowledgments

This work was supported in part by Deutsche Forschungsgemeinschaft Grant SFB 732. We thank Helmut Schmid for his comments and for support of BitPar.

References

[1] Fraser, Alexander; Wang, Renjing; Schütze, Hinrich. 2009. Rich bitext projection features for parse reranking. In EACL.

[2] Burkett, David; Klein, Dan. 2008. Two Languages are Better than One (for Syntactic Parsing). In EMNLP.

[3] Fossum, Victoria; Knight, Kevin. 2008. Using Bilingual Chinese-English Word Alignments to Resolve PP-attachment Ambiguity in English. In AMTA.

[4] Huang, Liang; Jiang, Wenbin; Liu, Qun. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. In EMNLP.

[5] McClosky, David; Charniak, Eugene; Johnson, Mark. 2006. Effective self-training for parsing. In NAACL.

[6] Collins, Michael. 2000. Discriminative Reranking for Natural Language Parsing. In ICML.

[7] Riezler, Stefan; King, Tracy; Kaplan, Ron; Crouch, Dick; Maxwell John; Johnson, Mark. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In ACL.

[8] Schmid, Helmut. 2004. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In COLING.

[9] Koehn, Philipp. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In MT Summit.

[10] Och, Franz. 2003. Minimum Error Rate Training in Statistical Machine Translation. In ACL.

[11] Och, Franz; Ney, Hermann. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29:1, 19-51.

[12] Koehn, Philipp; Och, Franz; Marcu, Daniel. 2003. Statistical Phrase-Based Translation. In HLT-NAACL.

[13] Brown, Peter; Della Pietra, Stephen; Della Pietra, Vincent; Mercer, Robert. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19:2, 263-311.