PARALLEL LFG GRAMMARS ON PARALLEL CORPORA: A BASE FOR PRACTICAL TRIANGULATION

Gerlof Bouma, Jonas Kuhn,
Bettina Schrader and Kathrin Spreyer
Universität Potsdam

Proceedings of the LFG08 Conference

Miriam Butt and Tracy Holloway King (Editors)

2008

CSLI Publications

http://csli-publications.stanford.edu/

Abstract

This paper presents an approach to annotation projection in a multi-parallel corpus, that is, a collection of translated texts in more than two languages. Existing analysis tools, like the LFG grammars from the ParGram project, are applied to two of the languages in the corpus and the resulting annotation is projected to a third language, taking advantage of the largely parallel character of f-structure. The third language can be a low-resource language. The technique can thus be particularly beneficial for corpus-based (cross-) linguistic research.

We discuss a number of ways to realize automatic corpus annotation based on multi-source projection, including direct projection and approaches with an additional generalization step that employs machine learning techniques. We present a series of detailed experiments for a sample annotation task, verb argument identification, using the German and English ParGram grammars for projection to Dutch and maximum entropy models for learning generalizations.

1 Introduction

With the rising prominence of corpus-based linguistics and linguistically grounded language technology, the demand for annotated corpora or wide-coverage tools that will add annotation automatically is increasing. However, the development of the necessary resources (through direct engineering, or indirectly through manual annotation of training data for machine learning techniques) is complicated and time-consuming, and, especially for low-density languages, the associated costs may be prohibitive. As a possible means of getting around this problem, researchers have investigated techniques of *annotation projection* (Yarowsky and Ngai, 2001; Yarowsky et al., 2001): annotation in a text in one language is transferred to a parallel text in a second language. This way, the development of resources in a language can benefit from existing resources in another language.

The research presented in this paper is situated in SFB 632, a large collaborative research programme that investigates the linguistic realization of information structure across languages. Corpus-supported research plays an important role in this cross-linguistically oriented programme and thus there is a need for medium-sized to large annotated corpora for many languages. Moreover, specialized linguistic research into information structure will often require annotations not included in standard treebanks: ways of providing additional annotations for more than one language would facilitate cross-linguistic research significantly. In this paper, we will therefore further investigate the method of automatically projecting

[†]The research reported in this paper has been supported by the German Research Foundation DFG, as part of SFB (Sonderforschungsbereich) 632 "Information structure: the linguistic means for structuring utterances, sentences and texts", University of Potsdam/Humboldt University at Berlin; `http://www.sfb632.uni-potsdam.de/`. The research was conducted in project D4 "Methods for interactive linguistic corpus analysis" (Principal Investigator: Kuhn).

annotation from better-studied to less-studied languages to create the required resources. Collections of translated text, parallel corpora, can be obtained relatively easily, even for low-resource languages. In particular, we investigate an extension of the annotation projection idea: Instead of projecting from a single source language, we can combine information from parallel texts in several languages (so called *multi-parallel* texts) to induce an annotation for the target text.

It is obvious that the quality of the projected annotation depends crucially on that of the source-side annotation; so annotation projection presupposes the availability of reliable wide-coverage tools for source language annotation. Multi-parallel annotation projection even requires that annotation (tools) exist for *several* languages and that the parallel annotations be comparable. In the form of the LFG grammars from the Parallel Grammar (ParGram) project (Butt et al., 2002), high-quality broad-coverage analysis tools are available for a number of languages, providing an excellent starting point. LFG's f-structure offers a level of analysis that exhibits great parallelism between languages and is thus suitable for projection, be it directly or in the form of more theory-neutral dependency structures derived from them. Moreover, the cross-linguistic stability of the analyses produced by different ParGram grammars is increased by the use of a carefully controlled common framework for grammar development.[1] The multi-source annotation projection approach is thus also an interesting new context of application for the ParGram grammars.

The rest of this paper is structured as follows. Section 2 describes multi-parallel annotation projection and its background. Then, to make the proposal more concrete and to be able to empirically study various aspects of multi-source annotation projection, Sections 3–5 describe our investigations of an example task that we approach by means of annotation projection. Section 3 introduces this task – *argument identification* – and discusses different ways in which multi-parallel corpora may be used in this task. After that, technical details of our implementation are given in Section 4. Finally, Section 5 gives experimental results of using consensus projection in various ways in the argument identification task. We offer a short conclusion and outlook in Section 6.

2 Multi-parallel annotation projection

The idea of exploiting parallel texts and cross-lingual parallelism to transfer existing annotations in one language to a new language first was brought forward by Yarowsky and Ngai (2001) and Yarowsky et al. (2001), who applied it to part-of-speech tagging, morphological analysis and NP bracketing. Their method of

[1] In contrast, when using standard treebank-trained parsers for source-side annotation (each based on the major available treebank for a language), one has to deal with considerably more, purely technical mismatches. For instance, the standard dependency treebank resources for German and English differ with respect to whether the highest verb or the complementizer is the head of a subordinate clause.

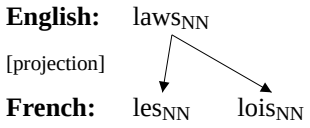


Figure 1: Incorrect POS-tag projection on the basis of 1–n alignment

annotation projection has been applied to a wide range of annotations, including dependency parsing (Hwa et al., 2005) and role semantic analysis (Padó and Lapata, 2005).

In these studies, annotation is projected from one language (the source language) to another (the target language). There are several circumstances in which annotation projection may be problematic. First, it may be that the source language does not make distinctions that the target language does make. In the context of projecting POS-tags, there may for instance be an important target-language distinction between prepositions and subordinating conjunctions, which the source language tag set may not make (as is the case for the Penn Treebank tag set (Marcus et al., 1993)). Similar situations may arise with regard to the adverb/adjective distinction, etc. This can to a certain extent be avoided by projecting a sufficiently general annotation. Post-processing the target annotation may also improve this situation somewhat. Secondly, the word alignment can be problematic if several words in one language are allowed to align to one in the other. Consider the alignment and projection in Figure 1, taken from Yarowsky et al. (2001). Since the English bare plural *laws*, with POS-tag NN (noun) is aligned with the French two word definite NP *les lois* 'the laws', naive annotation projection will assign the NN tag to both words. However, this is clearly only correct for *lois*. Yarowsky et al. (2001) solve this problem, too, by post-processing the target annotation.

A third source of problems for annotation projection may be the quality of the alignment itself. Given sentence level alignment, word level alignment for large corpora can be induced automatically (Brown et al., 1993). The quality of the resulting alignment is good enough to be used in a wide range of applications – most prominently in statistical machine translation. Nevertheless the alignment will contain many errors.

Each of the three problems can be looked upon as an instance of having too little information to correctly project annotation. In the case of a target annotation that is richer than the source, this is clear. Secondly, incorrect projection of annotation to a word that is part of a 1–*n* alignment (like assigning NN to *les* in Figure 1) could be avoided if we had some information about which alignments in a one-to-many configuration we should use and which not. Finally, the impact of noisy word alignment would be reduced if we had some information that would help us to recognize and filter out the noise from the true alignments.

The post-processing common in annotation projection work can be seen as an attempt to add this extra information. As an example of how effective this can be, we can take the work of Hwa et al. (2002) who project dependency structure

from English to Chinese. They report poor performance when simply projecting dependency structure (f-score 38.1; see footnote 5 for an explanation of f-score), but dramatically improve annotation quality (f-score 67.3) by applying transformations to the projected annotation based on independent knowledge of Chinese syntax.

Instead of manually inputting the extra information needed to improve the quality of the projected annotation, we propose to rely on a third (or fourth, etc.) parallel text as a source of information. For instance, target language details that are lacking in the annotation of one source language could well be present in the annotation of a second source language or they could be derivable from the combination of two projected annotations. Comparing multiple single source annotations may also tell us when we should be confident about an annotation (for instance, when the projected annotations agree) or when we are better off ignoring it. This offers possible solutions to the problem of not knowing which path in a 1–*n* alignment to use and the problem of not being able to tell noise from signal. Of course, using extra languages is not a watertight solution to the problems sketched above. It is in principle possible that all of the source languages project exactly the same wrong annotation, so that combining annotations does not help us at all. Still, we expect that by using an extra language this scenario becomes less likely, thereby increasing the overall quality of the projected annotation.

The (possibly redundant) use of two or more sources has been discussed in various guises in quite different contexts. In Machine Translation, the old idea of *triangulation* (originally due to Martin Kay) is considered a helpful tool for disambiguating translational choices (Och and Ney, 2001; Cohn and Lapata, 2007). In the same paper that introduces annotation projection, Yarowsky et al. (2001) demonstrate that using multiple aligned corpora improves automatic induction of morphological analyzers in a target language. Finally, it has been pointed out that multi-parallel data can be naturally interpreted as different views on the same data (Callison-Burch and Osborne, 2003). It can thus be exploited in machine learning methods that rely on having different views of the same data, such as co-training (Blum and Mitchell, 1998) and weakly supervised versions thereof (Hwa et al., 2003).

3 Annotation projection applied: Argument identification

To give a more concrete picture of the various aspects of multi-parallel annotation projection, the rest of the paper will lay out a projection approach to the *argument identification* task. This task consists of finding, for a given verb, the head words of its arguments. An example annotation is given in in (1), where each word is classified as an argument or non-argument of the given head verb *stellen* ('compose').[2,3]

[2] A given word can be a non-argument in several ways: It may be a modifier of the head word under consideration (like *morgen*, 'tomorrow'), further embedded inside an argument (*nieuwe*, 'new'), higher up than the head word in the embedding hierarchy (*als*, 'if'), or only indirectly related.

[3] Were needed, we abbreviate languages as follows: Dutch *dut*, English *eng*, German *ger*.

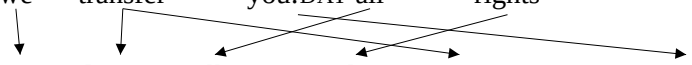
(1) Als$_{\text{non-arg}}$ wij$_{\text{arg}}$ morgen$_{\text{non-arg}}$ nieuwe$_{\text{non-arg}}$ regels$_{\text{arg}}$ **stellen**$_{\text{hd}}$ **...** (dut)
if we tomorrow new rules compose
'If we make new rules tomorrow...'

Annotation like this is useful in corpus investigations of argument frames or, especially when enriched with grammatical function labels, principles of argument ordering – an area that is directly relevant to the study of information structure, the topic of our larger research programme. At the same time, the task is conceptually simple enough to serve as a demonstration and to allow us to concentrate on the methodology. Since LFG's PRED values point to the lexical heads of f-structures and there is a list of argument functions, the relevant information is also directly available from an LFG parse.

3.1 Single source projection

In the most direct realization of an annotation projection approach to argument identification, we simply transfer for each word in the source language its argument status to aligned words in the target language. For instance, in (2), the German annotation and the word alignment are given and used to create the annotation of the Dutch sentence for the verb *over|dragen* (the verb particle is treated as a non-argument).

(2) Wir$_{\text{arg}}$ **übertragen**$_{\text{hd}}$ Ihnen$_{\text{arg}}$ alle$_{\text{non-arg}}$ Rechte$_{\text{arg}}$ (ger)
we transfer you.DAT all rights



Wij$_{\text{arg}}$ **dragen**$_{\text{hd}}$ alle$_{\text{non-arg}}$ rechten$_{\text{arg}}$ over$_{\text{non-arg}}$ aan$_{\text{non-arg}}$ u$_{\text{arg}}$ (dut)
we transfer all rights VPART to you

Argument status annotation of the source language can be created by parsing the source corpus and then labelling each word in a sentence whether it is the head of an argument of a selected verb in the sentence. Word alignment can also be induced automatically, given that we have a sufficiently large, sentence aligned corpus.

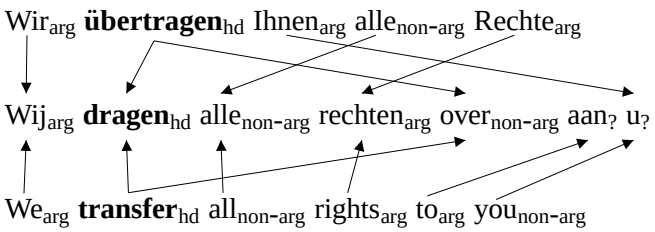
If we automatically word align the Europarl corpus, use the German and English LFG grammars to parse part of it in the respective languages, and then extract argument status from the LFG analyses, we can project this information from German to Dutch or from English to Dutch. We can compare this projected annotation to a manual annotation of the same corpus (Section 4 for details), which gives us the quality results of Table 1. Projecting from German or English to Dutch, we find about half of all arguments that are in the corpus (*recall*), and about half of the words that we project to be arguments are indeed arguments (*precision*). According to the manually annotated gold standard, about every tenth word is an argument. This means that projecting argument status from, say, German offers an improvement in precision over just picking random words from 10.1% to 52.2%.

	Precision	Recall	F-score
German → Dutch	52.2	52.9	52.6
English → Dutch	54.3	48.8	51.4
consensus	74.6	34.1	46.8

Table 1: Single source raw projection (top) and German/English consensus projection (bottom) for the argument identification task.[5]

3.2 Multi-source projection

As mentioned in the previous section, projection based on a single source language is likely to suffer quite immediately from problems like annotation mismatch, multi-word alignment, or alignment errors. By moving to multi-source projection, we hope to systematically increase the precision of the projection without using any language- and task-specific rules and heuristics. We combine projections in the following way: Only labels that would be assigned consistently by single source projection from two (or more) languages are actually assigned to the target language words. We shall refer to the result as a *consensus projection*. The German/English-consensus annotation for (2) is in (3):

(3)


Note that we have a third category now, indicated with '?'. This label is assigned when the two sources disagree. In (3), English projects argument status to the preposition *aan*, as English uses an oblique complement for the verb *transfer* just like Dutch does for *over*|*dragen*. The German counterpart, however, uses the dative and thus German projects argument status to the pronoun *u*. The result is that both words in the PP are labelled '?'.

Going back to our system that involves Europarl and LFG parses for the source languages, Table 1 shows that consensus projection indeed results in a comparatively high precision annotation. Recall and precision refer to those words labelled as arguments, which means that in these statistics the difference between

[5]Precision is a measure of correctness, in our case it is the number of words pairs that the system correctly classifies as argument-head pairs divided by the number of all word pairs that the system proposes. Recall is a measure of coverage, in our case the number of word pairs that the system correctly proposes divided by the number of actual head-argument pairs in the text. There is typically a trade off between precision and recall. F-score summarizes the overall system performance by taking the harmonic mean of the two measures: $f = 2 \times (\text{precision} \times \text{recall}) / (\text{precision} + \text{recall})$.

non-arguments and '?'s is ignored.[6] Of course, there is a precision/recall trade-off, but high precision can be very useful in a certain type of explorative linguistic corpus research, where the goal is to find some typical examples of a rare phenomenon in a large corpus, without requiring exhaustivity or representative samples. Furthermore, the high-precision results may be a good basis for machine learning of generalizations, which we will come to next.

3.3 Beyond raw projection

So far, annotation projection has been a completely deterministic process, which has not gone beyond the simple mechanism of projecting information over word alignments. Let us refer to this as raw projection, a term which is agnostic about the number of sources for the projection. We will try to improve the usefulness and/or quality of projection by using raw projection data in two ways.

Target language classifiers Attractive an idea as it may be, raw projection – by design – has a great disadvantage: In the end, we are tied to the parallel corpus. Hwa et al. (2005) use projection annotation to create training data for a statistical parser that itself does not rely on the parallel corpus. To explore this option, we have used machine learning techniques to construct an argument status classifier on the basis of data that we get from raw projection. An important empirical question is whether using consensus data has any advantages over the alternatives: training on single source projected data, or training on a small set of manually annotated data. Furthermore, since the motivation for annotation projection is to avoid time-intensive and costly manual labour, we shall compare the effectiveness of projected annotation (i.e., using high-quantity/medium-quality data) with manual annotation that takes about a day to create (low-quantity/high-quality data). Sections 5.1 and 5.2 discuss the results of these studies.

Using more information in the parallel data As it stands, we use only a fraction of the information that is available about the source languages in our raw projection method. The LFG parses give us much more information about the source than just argument-head relations. For instance, we also have information such as part-of-speech, case, finiteness, and agreement features for the words in the parsed source language sentences. We could exploit this information in a machine learning step that follows the raw projection step, by projecting the relevant features to the target language words. Although this move does not free us from the parallel corpus, one might hope that the quality of annotation improves compared to raw consensus projection, for instance by improving recall without sacrificing too much precision. An additional interesting question is how the resulting models compare to target language classifiers that use richer information about the target language, that is: can we replace target language information with source language

[6] The '?'-words would form an obvious starting point for adding some heuristics to further improve the projected annotation. However, in the present paper we focus on fully general, non-heuristic techniques.

information without losing annotation quality? Section 5.3 gives experimental results that answer these questions.

4 Details of the implementation

Having given a high level overview of the projection of argument status and the effects of moving from single source projection to multi-source consensus projection, we now describe in detail how we obtain projected annotations and stand-alone target language classifiers from the plain parallel corpus.

For the results reported in the previous section, and in the experiments of the next section, we use the Dutch, English, and German parts of the Europarl corpus (Koehn, 2005). Europarl consists of translations of the proceedings of the European Parliament in 11 languages ($\sim$30 million words in $\sim$1 million sentences per language). Dutch will serve as the target language in the experiments presented here. The use of Dutch as the target language instead of an actual low-density language is motivated in these exploratory stages: We are free to play around with the amount of resources we assume to be available in the target language (for instance, we can choose to POS-tag the target corpus or not), we have easy access to linguistic expertise in the target language that will help us evaluate the results of the projection, and, finally, we can use existing tools (like parsers) to simulate a target-language expert in the experiments that rely on input from such an expert.

To get from the plain parallel corpus of source and target language texts to annotated target language texts, one takes the following steps:[7] (1) The parallel target and source texts are tokenized, split into sentences, a sentence alignment is computed, which is then used as the basis for bootstrapping a statistical word alignment between the target and source language words in the sentences. (2) The source language corpora are parsed and the relevant information (most prominently head-argument relations) is extracted from the analyses. (3) The word alignments are used to project the head-argument relations from the source sentence(s) to the target sentence. The exact nature of the projected annotations depends on whether we choose to do single source or multi-source projection. For the experiments that involve machine learning, another step is added: (4) Using the projected data as training data, we train a binary argument status classifier. In one of the configurations, the classifier can refer to additional information from the source languages

[7] Various of the steps described go along with a potential loss of data. For instance, if the sentence alignment skips a sentence for one of the language pairs, the sentence cannot be used, even if it is included in the other language pair; parse failures for either of the source languages render the complete sentence triple unusable; free translations or errors in the automatic word alignment may make it impossible to identify candidate head words in the target language. For our experimental work, we did not attempt to minimize loss of data (which in some cases may not be too hard to do), since we had a sufficient amount of raw data to start with. From a potential set of 300k German-English-Dutch sentence triples submitted to the pipeline, we ended up with 52k usable triples. Note that processing is completely automatic, so no human resources are wasted. When working with a smaller corpus, the process may have to be optimized however.

that has not been used in raw projection. The details of these steps and our evaluation scheme in the following subsections.

4.1 Preprocessing

We used the sentence-aligned form of the Europarl corpus (Koehn, 2005). German–Dutch and English–Dutch word alignments were obtained with the GIZA++ tool (Och and Ney, 2003). The IMS TreeTagger (Schmid, 1994) was used for POS-tagging of the three languages. The POS-tags for Dutch are only used by some of the target language classifiers in the machine learning step (details in Section 4.4).

We parsed the German and English portions of the corpus with the ParGram LFG grammars running in the XLE environment (Crouch et al., 2007). Figure 2 shows how the information from the parses (LFG f-structures) flows from the German and English side to form a consensus annotation of a Dutch clause. As can be seen there, the German and English f-structures in this example are largely parallel. The only structural difference lies in the complexity of the direct object. This is in spite of the fact that the phrase structures are not very alike at all.

4.2 Feature extraction

Since our projection task is defined at the level of words and word alignments, we need to transfer the grammatical information encoded in the f-structure to the tokens of the parsed string. This information of course includes grammatical dependencies, but also a host of other information that the ParGram grammars use in parsing. This step of flattening down the nested f-structures to properties of words in the string (that is, c-structure terminals) is labelled '[feature extraction]' in Figure 2. The extraction of all relevant features is implemented as a set of rewrite rules which are executed by XLE's transfer system (using the `extract` command; see Section 'Transfer' in Crouch et al. (2007)). Note that the information extracted from f-structure is always relative to a particular head word in the sentence. Hence, for all other words in the sentence, the f-structure path under which they are embedded with respect to the head word (if there is such a path) can be uniquely specified. Henceforth, we call this specification the *path* feature of the word. Figure 2 shows for each token the extracted path features with respect to the head verbs *erlassen* and *make* in German and English, respectively.

The rich information encoded in the deep LFG grammars enables the XLE parser to augment the preliminary tokenization provided in the input. These modifications chiefly concern compound words and multi-word expressions. However, since the word alignment is defined on the Europarl tokenization, we need an additional mapping step after the mapping of information from f-structures to c-structure terminals. For instance, in the German '[retokenization]'-step in Figure 2, we can see that the compound *Rechts|vorschriften* 'regulations' has been deconstructed by the parser. The compound in the Europarl tokenization will receive the features extracted for the head according to XLE.

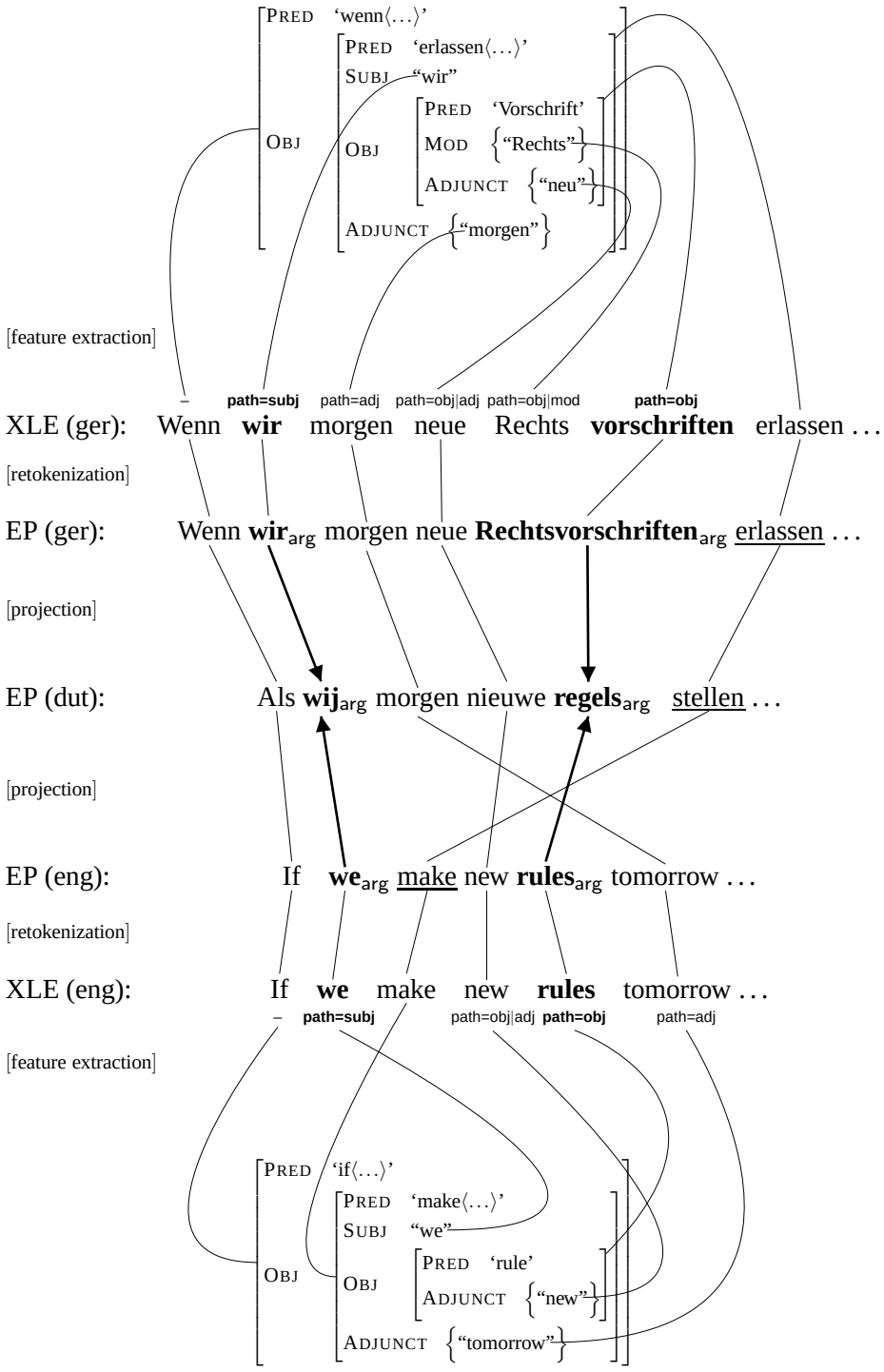


Figure 2: Multi-source consensus projection of head-argument relations from German and English f-structures to Dutch. Dutch *wij* and *regels* align with arguments of *erlassen* in German and *make* in English, fulfilling the consensus criterion.

4.3 Annotation projection

The extracted path features can be directly used to generate the source language annotation (i.e., assigning the label *arg* or *non-arg* to each word, relative to a particular target verb): A word d is an argument of the given verb h when there is an extracted feature $path(h, d, gf)$, and gf is one of SUBJ, OBJ, OBJ-TH, XCOMP, XCOMP-PRED, VCOMP, COMP, or any of the variants of OBL.

Having established the relevant information on the source language tokens in the word-aligned parallel corpus, we can determine whether a target language word w is an argument of a given target language verb h under *single source projection*: A target language word w is an argument of h iff there are aligned source language words w' and h' such that w' is an argument of h'.

The *consensus projection* on the basis of two languages is defined in terms of two single source projections. Target word w is an argument of h when the two single source projections agree it is and w is not an argument of h when the two single source projections agree it is not. Otherwise, w,h is annotated with a '?'.

4.4 Machine Learning

We used the MegaM software package[8] to train maximum entropy (maxent) binary classifiers for the argument identification task. The classifiers make use of features relating to the head word and the candidate argument, and features relating to the context. However, the classifiers do not take into account any information about the argument status of other words than the candidate. We used the default settings of MegaM and did not try to optimize parameter settings such as those concerning penalization.

The features used by the maxent classifiers fall into four categories: lexical, contextual, alignment geometry, and projected features. *Lexical features* include the surface form, as well as the lemma and POS-tag of the candidate argument (if available). *Contextual features* are sentence length, position and distance between the head and the candidate, POS-tags of adjacent words, and intervening complementizers, verbs, and punctuation. *Alignment geometry features* encode information about the word alignment, such as the number of words the candidate is aligned to (as an indication of uncertainty in the word alignment). *Projected features* are used in Section 5.3, where we explore the use of source language information in the post-raw-projection stage. Instead of providing the classifiers with Dutch POS-tag and lemma information (as in the feature set described as 'dut features'), the Dutch words are here marked with feature information projected from the aligned German and English tokens, including NP form, person, number, tense, voice, aspect, verb type and clause type (this feature set is referred to as 'ger+eng+surface dut features'). Finally, the feature space also includes selected conjunctions of atomic features.

[8] See `http://www.cs.utah.edu/~hal/megam/`

	Precision	Recall	F-score
3 nearest nominals	37.0	50.6	42.7
Alpino	87.1	92.5	89.7
	Averaged performance		
Random baseline (hypothesized)	10.1	-	-
Classifier trained on 100k data points with expert annotation, dut features (50 runs)	73.4	53.9	62.4

Table 2: Upper and lower comparison points for the argument identification task.

Some of the machine learning experiments do not use raw projection data to train the classifiers, but rather expert annotated data. For these experiments, we make use of the Alpino parser for Dutch (Malouf and van Noord, 2004) to simulate human input. We use simulated expert input rather than actual human input to be able to repeatedly train the same models on randomly selected training sets. The amount of expert annotated training data is kept around 200 verbal heads (or 4k data points), an amount we estimate can be manually annotated in a matter of hours.

4.5 Evaluation

We evaluate the annotation produced by raw projection or by one of the machine learned classifiers by comparing it to a small gold standard annotated by a linguistically trained native speaker of Dutch. The annotation largely follows the guidelines of the spoken Dutch corpus CGN (Hoekstra et al., 2003), with the important difference that not whole phrases, but only the head words of phrases are annotated as arguments. The gold standard consists of 240 verbal heads in 222 sentences, giving a total of 4756 data points.[9]

Table 2 describes the gold standard in terms of the performance of several reference approaches to the argument identification task. For instance, a simple heuristic that assigns argument status to the three nominals that are closest to the selected head achieves precision of 37.0% and recall almost as high as the single source projections of Section 3. The Alpino parser can be used for argument identification by simply extracting the relevant verb/argument pairs from the full parsing output. It performs very well when tested against the gold standard annotation. This justifies its use in the simulation of expert annotation. A non-deterministic baseline that would randomly assign argument status to words, would average a precision that is equal to the proportion of argument-head pairs in the gold standard. In our case, this is 10.1%.

In the evaluation of machine learned classifiers, it is important to try to rule out the possibility that the observed results are simply due to the fact that the training

[9] We count a data point for each candidate pair w, h. This number is typically greater than the total number of words in the selected subcorpus, since the same word w may be paired with different hs in the sentence.

data is very much like or unlike the evaluation data. We therefore report average results over 50 runs of machine learning on randomly selected sets of training data. About these averages it is important to realize that they need not correspond to an actual run, nor is it guaranteed that one can train a model that performs exactly like that. Apart from the averages, it is also instructive to inspect the variation of performance between runs: of two equally performing systems, one should prefer one that shows little variation between test runs, as this system is more likely to perform similarly on unseen data in the future.

As an upper limit for the machine learned classifiers based on our representations and our set of learning features, Table 2 reports a classifier that has access to all and only Dutch features, which was trained on 100k data points that were annotated by our simulated expert. The annotation quality on the evaluation data is considerably lower than that of a carefully designed full parser like Alpino. This comparison shows that despite the conceptual simplicity of the argument identification task, it is a very hard classification task to acquire in isolation, using machine learning techniques. Some of the difficulties are intuitively clear, as argument identification is cast as an extremely local decision. Knowing whether or not *cheese* in *I don't like cheese crackers* has been classified as an argument of *like* should influence the argument identification decision for *crackers*, but the classifier is ignorant of this.[10] Contrary to the purely local application of our sample classifier, any full parser will build smaller units (phrasal constituents or dependency subgraphs), e.g., [*cheese crackers*] and incorporate valency knowledge. As our goal is to investigate what relative improvements can be obtained with various annotation projection techniques, the hardness of the sample task is not problematic in itself – to the contrary, it is interesting to study the techniques specifically for a hard task.[11] At the same time, one has to be aware that the results we obtain are likely to depend on the choice of the sample task.

5 Experimental results

We will now turn to the experimental investigation of the three questions that were raised in Section 3.3. To start, we investigate whether the high-precision annotation that is the result of consensus projection is of use in training a target language classifier for the argument identification task (Section 5.1). We then compare the utility of high-quantity/medium-quality projection data to low-quantity/high-quality data (Section 5.2). Finally, we look into the question of whether we can replace some target language information in the classifiers with parallel source language information (Section 5.3).

[10] With contextual POS tag features, which are only included in the richest feature set 'dut features', it may be possible to learn typical noun-noun compound contexts, so the resulting classifier may have a preference for the correct reading of the *cheese crackers* example.

[11] Practical uses of the projection technique in the context of linguistic research on information structure can be expected to involve similarly hard classification tasks; so, a realistic application has to employ some interactive, semi-automatic regime.

5.1 Target language classifiers trained on consensus projection

We have seen in Section 3.2 that, compared to single-source projection, consensus projection leads to high-precision annotation. However, since consensus projection is the intersection of two single-source projections, the increase in precision comes with a considerable loss of recall. The first question we look at is whether the resulting high precision annotation can be used as training data for a generalizing target language classifier, that is, a classifier that is independent of information from the parallel corpus. To this end, we compare average performance of three systems. In each system, the classifier uses Dutch lemma and POS-tag information to make predictions. The systems differ in whether they are trained on 100k data points with labels projected from German, from English, or from both languages.

The results of these experiments are shown in Table 3,[12] together with the repeated results of the various raw projection methods. The results show that the generalization step on average ameliorates the recall problem observed for raw consensus projection (from 34.1 to 39.9). The improved f-score (from 46.8 to 49.5) shows that the resulting classifier strikes a better balance between precision and recall than raw consensus projection does.

If we compare the different classifiers, we can see that the consensus data trained classifiers (f-score 49.5) offer overall improvement over the single source data trained classifiers (f-scores 45.6 and 44.6 respectively).[13] We draw the conclusion that by using consensus data we are able to induce higher quality target-language classifiers compared to single-source projection.

In addition to the increase in average performance, the boxplots clearly indicate that the variation in recall and precision of the classifiers is less for the consensus data trained classifiers. We interpret the increased stability of the classifiers based on consensus data as symptomatic for the noise filtering function of the consensus projection, speculated upon in Section 2.

In all, these first series of experiments provide evidence that for tasks like our example task, consensus projection offers an advantage over single source annotation projection for the induction of target language classifiers.

[12] Each projection+generalization method was run 50 times using randomly selected Europarl sentences and tested each time on the held-out gold standard. For instance, there are 50 precision measurements for 'consensus, dut features': $58.2, 59.1, \ldots, 64.2, 64.3, 64.4, 64.4, \ldots, 70.4, 70.7$ (sorted). These outcomes are summarized by the average (median) and boxplots. All boxplots are on a scale 0–100 points. The boxplot whiskers indicate the $1.5\times$inter-quartile-range area around the central 50 percent of the data. Outliers are plotted as dots. The systems without a generalization step are deterministic, so the reported performance measures are not averages as such. For ease of comparison, the corresponding 'boxplots' are drawn as medians only.

[13] The consensus data trained classifiers offer statistically significant improvement over the ones trained on data projected from German in terms of average precision (median $+5.1$, $p < .001$), average recall ($+1.6$, $p = .023$) and average f-score ($+3.9$, $p < .001$), and over those trained on data projected from English in terms of average recall ($+4.4$,$p < .001$) and average f-score ($+4.9$, $p < .001$) although not in terms of average precision ($+0.8$, $p = .169$).

Significance testing is done with approximate randomization testing of the median. The p-values are based on 50k random resamplings of the pooled evaluation data of two systems. See Yeh (2000) for references and recommendations.

Projection	Generalization	Average	Distribution
		Precision	
ger→dut	(none)	52.2	
	dut features	59.3	
eng→dut	(none)	54.3	
	dut features	63.6	
consensus	(none)	74.6	
	dut features	64.4	
		Recall	
ger→dut	(none)	52.9	
	dut features	38.3	
eng→dut	(none)	48.8	
	dut features	35.6	
consensus	(none)	34.1	
	dut features	39.9	
		F-score	
ger→dut	(none)	52.6	
	dut features	45.6	
eng→dut	(none)	51.4	
	dut features	44.6	
consensus	(none)	46.8	
	dut features	49.5	

Note: Raw projection, labelled '(none)', is tied to parallel corpus data, whereas the generalized 'dut features' classifiers can be applied to arbitrary text.

Table 3: Performance of raw projection and target-language internal classifiers. Also see footnote 12 for explanation.

5.2 Consensus projection instead of low-quantity/high-quality data

Since the purpose of annotation projection is to avoid time- and cost-intensive manual efforts, a relevant question to ask is how a consensus projection based system fares against a system that relies on a modest amount of manually annotated data, as such a system would be a practical alternative. Table 4 compares the results of training a classifier on 100k data points of consensus data and training it on 4k expert annotated data points. The table shows that using expert data results in considerably higher average precision, but that it makes no difference in average recall. In terms of f-score, there is a small overall advantage in using expert annotated data.[14]

Two things need to be pointed out about this comparison. First, the picture in

[14] 100k consensus vs 4k expert: average precision $+7.3$, $p < .001$, average recall -0.9, $p = .107$, and average f-score $+1.5$, $p = .002$.

Projection	Generalization	Precision Average	Precision Distribution
consensus	dut features	64.4	
(expert)	dut features	71.7	
		Recall	
consensus	dut features	39.9	
(expert)	dut features	39.0	
		F-score	
consensus	dut features	49.5	
(expert)	dut features	50.8	

Table 4: Training on 100k consensus data vs. 4k expert data.

Table 4 shows smaller variation for the consensus trained models on all fronts. This is likely to be an effect of using much more data and using data that is systematically filtered because of the consensus requirement. In a recall-oriented approach, it is thus preferential to use a lot of consensus data over a modest amount of expert data: The reduced stability of the latter means that a particular instance may in fact perform much worse than average. Recall-oriented approaches are, for instance, useful for linguistic research in the semi-automatic set-up common in lexicography, which involves generating a candidate list of items that is then checked by a linguistic expert.

Secondly, the consensus approach has the advantage that, depending on the specific classification task, it may be possible to improve upon performance by simply adding more data, at merely the cost of more computation time.

5.3 Trading target language for source language information

The classifiers we examined in the previous section relied on the availability of some (admittedly fairly low level) analysis of the target level language, that is, lemma and POS-tag information. For a low-density language it may be the case that this level of analysis is not available. To compensate, one may look at information that comes from the parallel corpus to replace the target language information. The price one pays for this move is that the final classifier remains dependent on being applied in the context of a multi-parallel corpus.[15]

Table 5 gives an overview of the performance of systems that incorporate multi-parallel information in the generalization step: Recall that the 'dut features' classifier has access to lemma and POS-tag information, which is missing from 'surface dut features'. The classifier 'ger+eng+surface dut features' can draw on projected

[15]The technique may of course still be of interest as a stepping stone in some bootstrapping cycle that reaches independence from the parallel corpus at a later stage.

Projection	Generalization	Precision Average	Precision Distribution
consensus	dut features	64.4	
	surface dut features	77.6	
	ger+eng+surface dut features	62.3	
		Recall	
consensus	dut features	39.9	
	surface dut features	20.6	
	ger+eng+surface dut features	38.3	
		F-score	
consensus	dut features	49.5	
	surface dut features	32.2	
	ger+eng+surface dut features	47.3	

Table 5: Impact of reduced target language information on performance.

morphosyntactic features from German and English, which may thus in part stand in for the lacking information. Only information about the verb and argument candidate itself is projected, so there is no stand-in for the contextual information included in 'dut features'.

We begin by noting that there is a considerable penalty in terms of recall and f-score if we withhold POS-tag and lemma information from the target-language internal classifier. Unexpectedly, however, precision increases to levels that on average lie even above consensus projection precision. The explanation for this is that the classifier that only relies on Dutch surface forms is very conservative. It learns for a limited number of surface forms that they are arguments. For instance, one will see that these classifiers always predict that *wij* 'we' and *ik* 'I' are arguments. This is correct: Dutch nominative pronouns are almost exclusively found in subject position. Under this strategy, high precision goes hand in hand with low recall. The fact that these models basically list specific cases also explains why there is such enormous variation in the precision of these models and relatively large variation in recall and f-score: The effectiveness of the listing approach relies directly on how well the training data resembles the testing data.

The use of morpho-syntactic features from German and English in addition to the Dutch surface features makes up for the loss in recall and f-score to a great extent. The resulting system still performs worse than a system with access to Dutch POS-tags and lemmata,[16] but the differences are modest. Whether the remaining differences are due to the lack of contextual features in 'ger+eng+surface dut features' needs to be investigated in future research.

[16] 'ger+eng+surface dut features' vs. 'dut features': average precision -2.1, $p < .001$, average recall -1.6, $p < .001$, average f-score -2.2, $p < .001$.

One may argue that if one has access to a parallel corpus anyway, one might as well use a raw projection method. Some of these even perform better than 'ger+eng+surface dut features': Table 1 shows that raw projection from German achieves f-score 52.6. However, classifiers do have the advantage that they can assign a confidence level to a classification. The results in Table 5 suggest that, if one needs classification confidence in a parallel corpus, one could replace target language information with source language information without sustaining too much of a hit in performance. One scenario we intend to explore in which this ability is relevant is so-called *active learning*. This involves iteratively improving a classifier by letting the machine learner select small amounts of training data to be annotated by an expert, on the basis of its own classification confidence. Nevertheless, we are aware that, as it is, projected feature information as a stand-in for target language POS-tags and lemmata is of a limited use.

6 Conclusion

We have presented an extension of annotation projection, in which we exploit multi-parallel corpora and use two (or more) languages as the source of projection. The source language annotation is automatic and based on the ParGram LFG grammars. These grammars are a very good basis for this technique, as they are designed to assign syntactic analyses to the two source language strings that are as parallel as is linguistically justified.

We illustrated and tested the technique for a sample task (verb argument identification), in which one has to decide whether a word is the head of an argument of a given verb. We have compared various ways of automatically annotating the target language in experiments with English and German as source languages and Dutch the target language, using Europarl data with standard statistical word alignments. The simplest automatic annotation method is direct, 'raw' projection of annotation from a single source, which yields f-scores of 52.6 (German to Dutch) and 51.4 (English to Dutch; see Table 1). Precision and recall are very balanced for single-source projection. The idea of multi-source projection can be implemented by relying only on the consensus of the two simpler projections. As expected, raw consensus projection reduces recall but results in high precision (74.6).

The argument identification task, which we chose for its conceptual simplicity, turned out to be a hard task to train an automatic classifier for. This is shown by the fact that even when a relatively large amount of expert-annotated training data is provided and a rich, reliable set of learning features is used, the classification quality is nowhere near the quality that can be reached by reading the argument identification decision off the output of a carefully designed full parser like the Alpino parser (Table 2). Since we were interested in ways of obtaining relative improvements by projection-informed techniques, a hard sample task is actually a good starting point. One should be aware, however, that details in the results may depend on our sample task.

Because the sample task is so hard, it is not trivial to train a stand-alone classifier for the target language without providing any expert information on the target language classification. However, the experiments reported in Section 5.1 show that with training data obtained from consensus projection, a significant improvement over training on single source projected data can be achieved (Table 3). The recall problem of raw consensus projection can be alleviated by generalizing over the projected data. Moreover, the resulting stand-alone classifier (average f-score of 49.5) is applicable outside the context of a parallel corpus too.

We further compared automatic classifiers based on consensus projected training data vs. small amounts of manually labelled training data (Section 5.2) and we tested to what degree source language information may replace target language POS-tag information – which may not be available for a low-resource language (Section 5.3). In both cases, the use of automatically obtained, multi-parallel projection information yielded performance only slightly inferior to the more resource intensive alternatives. A consistent pattern was that multi-parallel projection information helped to achieve high stability in classifier performance over training trials. This shows that the projection technique is less dependent on the contingent similarity between training and application data.

In future work, we plan to investigate projection of different types of annotation and to do an application test of semi-automatic versions of the technique in corpus-supported linguistic research and especially in an active learning setting. We are also planning to explore the use of more noise robust learning techniques. The local character of the argument identification decision, discussed briefly at the end of Section 4.5, may be typical for some annotation projection tasks. However in general, multi-parallel projection should be combined with more globally informed models. This is one of the most prominent goals of our ongoing work. For instance, we are now exploring full dependency parsing of the target language. This poses some interesting research questions, like what should be counted as consensus when the structural homomorphism across languages does not exhaust the entire candidate sentence.

References

Blum, A. and Mitchell, T. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the 1998 Conference on Computational Learning Theory*, pages 92–100.

Brown, P. E., Della Pietra, V. J., Della Pietra, S. A. and Mercer, R. L. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19(2), 263–311.

Butt, M., Dyvik, H., King, T. Holloway, Masuichi, H. and Rohrer, C. 2002. The Parallel Grammar Project. In *Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation*, pages 1–7.

Callison-Burch, C. and Osborne, M. 2003. Co-Training For Statistical Machine Translation. In *Proceedings of the 6th Annual CLUK Research Colloquium*.

Cohn, T. and Lapata, M. 2007. Machine Translation by Triangulation: Making Effective Use of Multi-Parallel Corpora. In *Proceedings of the 45th Annual Meeting of the ACL*, pages 728–735, Prague.

Crouch, D., Dalrymple, M., Kaplan, R., King, T., Maxwell, J., and Newman, P. 2007. XLE Documentation. Palo Alto Research Center. `http://www2.parc.com/isl/groups/nltt/xle/doc/`.

Hoekstra, H., Moortgat, M., Renmans, B., Schouppe, M., Schuurman, I. and van der Wouden, T. 2003. CGN Syntactische annotatie. `http://ww2.tst.inl.nl/images/stories/docs/syn_prot.pdf`.

Hwa, R., Osborne, M., Sarkar, A. and Steedman, M. 2003. Corrected Co-training for statistical parsers. In *Proceedings of the 20th International Conference on Machien learning*, Washington, D.C.

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C. and Kolak, O. 2005. Bootstrapping Parsers via Syntactic Projection across Parallel Texts. *Natural Language Engineering* 11(3), 311–325.

Hwa, R., Resnik, P., Weinberg, A. and Kolak, O. 2002. Evaluating Translational Correspondence using Annotation Projection. In *Proceedings of ACL-2002*, Philadelphia, PA.

Koehn, P. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the MT Summit 2005*.

Malouf, R. and van Noord, G. 2004. Wide Coverage Parsing with Stochastic Attribute Value Grammars. In *IJCNLP-04 Workshop Beyond Shallow Analyses - Formalisms and statistical modeling for deep analyses*.

Marcus, Mitchell P., Santorini, Beatrice and Marcinkiewicz, Mary Ann. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics* 19(2), 313–330.

Och, F. J. and Ney, H. 2001. Statistical Multi-Source Translation. In *MT Summit 2001*, pages 253–258, Santiago de Compostela, Spain.

Och, F. J. and Ney, H. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29(1), 19–51.

Padó, S. and Lapata, M. 2005. Cross-lingual projection of role-semantic information. In *Proceedings of HLT/EMNLP 2005*, Vancouver, BC.

Schmid, H. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, England.

Yarowsky, D. and Ngai, G. 2001. Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection across Aligned Corpora. In *Proceedings of NAACL-2001*, pages 200–207.

Yarowsky, D., Ngai, G. and Wicentowski, R. 2001. Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora. In *Proceedings of HLT 2001*.

Yeh, A. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of COLING 2000*, pages 947–953.