Experiments in Parallel-Text Based Grammar Induction

Jonas Kuhn
Department of Linguistics
The University of Texas at Austin
Austin, TX 78712
jonak@mail.utexas.edu

Abstract

This paper discusses the use of statistical word alignment over multiple parallel texts for the identification of string spans that cannot be constituents in one of the languages. This information is exploited in monolingual PCFG grammar induction for that language, within an augmented version of the inside-outside algorithm. Besides the aligned corpus, no other resources are required. We discuss an implemented system and present experimental results with an evaluation against the Penn Treebank.

1 Introduction

There have been a number of recent studies exploiting parallel corpora in bootstrapping of monolingual analysis tools. In the "information projection" approach (e.g., (Yarowsky and Ngai, 2001)), statistical word alignment is applied to a parallel corpus of English and some other language F for which no tagger/morphological analyzer/chunker etc. (henceforth simply: analysis tool) exists. A high-quality analysis tool is applied to the English text, and the statistical word alignment is used to project a (noisy) target annotation to the F version of the text. Robust learning techniques are then applied to bootstrap an analysis tool for F, using the annotations projected with high confidence as the initial training data. (Confidence of both the English analysis tool and the statistical word alignment is taken into account.) The results that have been achieved by this method are very encouraging.

Will the information projection approach also work for less shallow analysis tools, in particular full syntactic parsers? An obvious issue is that one does not expect the phrase structure representation of English (as produced by state-of-the-art treebank parsers) to carry over to less configurational languages. Therefore, (Hwa et al., 2002) extract a more language-independent dependency structure from the English parse as the basis for projection to Chinese. From the resulting (noisy) dependency treebank, a dependency parser is trained using the techniques of (Collins, 1999). (Hwa et al., 2002) report that the noise in the projected treebank is still a major challenge, suggesting that a future research focus should be on the filtering of (parts of) unreliable trees and statistical word alignment models sensitive to the syntactic projection framework.

Our hypothesis is that the quality of the resulting parser/grammar for language F can be significantly improved if the training method for the parser is changed to accomodate for training data which are in part unreliable. The experiments we report in this paper focus on a specific part of the problem: we replace standard treebank training with an Expectation-Maximization (EM) algorithm for PCFGs, augmented by weighting factors for the reliability of training data, following the approach of (Nigam et al., 2000), who apply it for EM training of a text classifier. The factors are only sensitive to the constituent/distituent (C/D) status of each span of the string in F (cp. (Klein and Manning, 2002)). The C/D status is derived from an aligned parallel corpus in a way discussed in section 2. We use the Europarl corpus (Koehn, 2002), and the statistical word alignment was performed with the GIZA++ toolkit (Al-Onaizan et al., 1999; Och and Ney, 2003).[1]

For the current experiments we assume no preexisting parser for any of the languages, contrary to the information projection scenario. While better absolute results could be expected using one or more parsers for the languages involved, we think that it is important to isolate the usefulness of exploiting just crosslinguistic word order divergences in order to obtain partial prior knowledge about the constituent structure of a language, which is then exploited in an EM learning approach (section 3). Not using a parser for some languages also makes it possible to compare various language pairs at the same level, and specifically, we can experiment with grammar induction for English exploiting various

[1] The software is available at http://www.isi.edu/~och/GIZA++.html

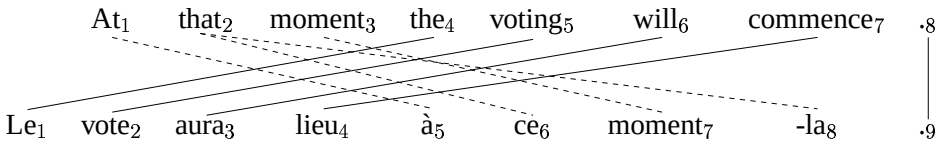


Figure 1: Alignment example

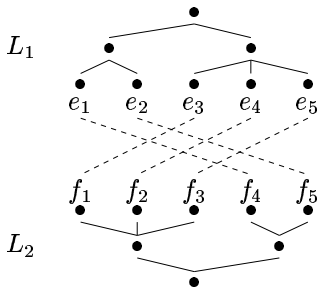
other languages. Indeed the focus of our initial experiments has been on English (section 4), which facilitates evaluation against a treebank (section 5).

2 Cross-language order divergences

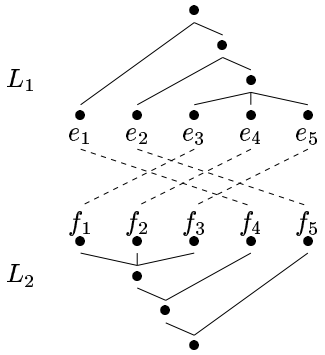
The English-French example in figure 1 gives a simple illustration of the partial information about constituency that a word-aligned parallel corpus may provide. The en bloc reversal of subsequences of words provides strong evidence that, for instance, [*moment the voting*] or [*aura lieu à ce*] do *not* form constituents.

At first sight it appears as if there is also clear evidence for [*at that moment*] forming a constituent, since it fully covers a substring that appears in a different position in French. Similarly for [*Le vote aura lieu*]. However, from the distribution of contiguous substrings alone we cannot distinguish between two the types of situations sketched in (1) and (2):

(1)



(2)



A string that is contiguous under projection, like e_1e_2 (1) may be a true constituent, but it may also be a non-constituent part of a larger constituent as in L_1 in (2).

Word blocks. Let us define the notion of a *word block* (as opposed to a phrase or constituent) induced by a word alignment to capture the relevant property of contiguousness under translation.[2] The alignments induced by GIZA++ (following the IBM models) are asymmetrical in that several words from L_2 may be aligned with one word in L_1, but not vice versa. So we can view a word alignment as a function α that maps each word in an L_1-sentence to a (possibly empty) subset of words from its translation in L_2. For example, in figure 1, $\alpha(\textit{voting}_5)$ =$\{\textit{vote}_2\}$, and $\alpha(\textit{that}_2)$ = $\{\textit{ce}_6, \textit{-la}_8\}$. Note that $\alpha(w_i) \cap \alpha(w_j) = \emptyset$ for $w_i \neq w_j$. The α-images of a sentence need not exhaust the words of the translation in L_2; however it is common to assume a special empty word NULL in each L_1-sentence, for which by definition $\alpha(\text{NULL})$ is the set of L_2-words not contained in any α-image of the overt words.

We now define an **α-induced block** (or α-block for short) as a substring $e_1 \ldots e_i$ of a sentence in L_1, such that the union over all α-images ($\bigcup_{1..i} \alpha(e_i)$) forms a contiguous substring in L_2, modulo the words from $\alpha(\text{NULL})$.

For example, $e_1e_2e_3$ in (1) (or (2)) is *not* an α-block since the union over its α-images is $\{f_1, f_4, f_5\}$ which do not form a contiguous string in L_2. The sequences e_3e_4 or $e_3e_4e_5$ are α-induced blocks.

Let us define a **maximal α-block** as an α-block $e_i \ldots e_j$, such that adding e_{i-1} at the beginning or e_{j+1} at the end is either (i) impossible (because it would lead to a non-block, or e_{i-1} or e_{j+1} do not exist as we are at the beginning or end of the string), or (ii) it would introduce a new crossing alignment

[2] The block notion we are defining in this section is indirectly related to the concept of a "phrase" in recent work in Statistical Machine Translation. (Koehn et al., 2003) show that exploiting all contiguous word blocks in phrase-based alignment is better than focusing on syntactic constituents only. In our context, we are interested in inducing syntactic constituents based on alignment information; given the observations from Statistical MT, it does not come as a surprise that there is no direct link from blocks to constituents. Our work can be seen as an attempt to zero in on the distinction between the concepts; we find that it is most useful to keep track of the *boundaries* between blocks.

(Wu, 1997) also includes a brief discussion of *crossing constraints* that can be derived from phrase structure correspondences.

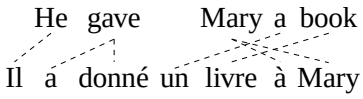
to the block.[3]

String e_3e_4 in (1) is not a maximal α-block, because $e_3e_4e_5$ is an α-block; but $e_3e_4e_5$ is maximal since e_5 is the final word of the sentence and $e_2e_3e_4e_5$ is a non-block.

We can now make the initial observation precise that (1) and (2) have the same block structure, but the constituent structures are different (and this is not due to an incorrect alignment). e_1e_2 is a maximal block in both cases, but while it is a constituent in (1), it isn't in (2).

We may call maximal blocks that contain only non-maximal blocks as substrings **first-order maximal α-blocks**. A maximal block that contains other maximal blocks as substrings is a **higher-order maximal α-block**. In (1) and (2), the complete string $e_1e_2e_3e_4e_5$ is a higher-order maximal block. Note that a higher-order maximal block may contain substrings which are non-blocks.

Higher-order maximal blocks may still be non-constituents as the following simple English-French example shows:

(3)


The three first-order maximal blocks in English are [*He gave*], [*Mary*], and [*a book*]. [*Mary a book*] is a higher-order maximal block, since its "projection" to French is contiguous, but it is not a constituent. (Note that the VP constituent *gave Mary a book* on the other hand is not a maximal block here.)

Block boundaries. Let us call the string position between two maximal blocks an **α-block boundary**.[4] In (1)/(2), the position between e_2 and e_3 is a block boundary.

We can now formulate the

(4) **Distituent hypothesis**

> If a substring of a sentence in language L_1 crosses a first-order α-block boundary (zone[5]), then it can only be a constituent of L_1 if it contains at least one of the two maximal α-blocks separated by that boundary in full.

This hypothesis makes it precise under which conditions we assume to have reliable negative evidence against a constituent. Even examples of complicated structural divergence from the classical MT literature tend not to pose counterexamples to the hypothesis, since it is so conservative. Projecting phrasal constituents from one language to another is problematic in cases of divergence, but projecting information about distituents is generally safe.

Mild divergences are best. As should be clear, the α-block-based approach relies on the occurrence of reorderings of constituents in translation. If two languages have the exact same structure (and no paraphrases whatsoever are used in translation), the approach does not gain any information from a parallel text. However, this situation does not occur realistically. If on the other hand, massive reordering occurs without preserving *any* contiguous subblocks, the approach cannot gain information either. The ideal situation is in the middleground, with a number of mid-sized blocks in most sentences. The table in figure 2 shows the distribution of sentences with n α-block boundaries based on the alignment of English and 7 other languages, for a sample of c. 3,000 sentences from the Europarl corpus. We can see that the occurrence of boundaries is in a range that should make it indeed useful.[6]

	L_2:						
n	de	el	es	fi	fr	it	sv
1	82.3%	76.7%	80.9%	70.2%	83.3%	82.9%	67.4%
2	73.5%	64.2%	74.0%	55.7%	76.0%	74.6%	58.0%
3	57.7%	50.4%	57.5%	39.3%	60.5%	60.7%	38.4%
4	47.9%	40.1%	50.9%	29.7%	53.3%	52.1%	31.3%
5	38.0%	30.6%	42.5%	21.5%	45.9%	42.0%	23.0%
6	28.7%	23.2%	33.4%	15.2%	36.1%	33.4%	15.2%
7	22.6%	17.9%	28.0%	10.2%	30.2%	26.6%	11.0%
8	17.0%	13.6%	22.4%	7.6%	24.4%	21.8%	8.0%
9	12.3%	10.3%	17.4%	5.4%	19.7%	17.3%	5.6%
10	9.5%	7.8%	13.7%	3.4%	16.3%	13.1%	4.1%

de: German; el: Greek; es: Spanish; fi: Finnish;
fr: French; it: Italian; sv: Swedish.

Figure 2: Proportion of sentences with $\geq n$ α-block boundaries for L_1: English

Zero fertility words. So far we have not addressed the effect of finding zero fertility words, i.e., words e_i from L_1 with $\alpha(e_i) = \emptyset$. Statistical word alignment makes frequent use of this mechanism. An actual example from our alignment is shown in figure 3. The English word *has* is treated as a zero fertility word. While we can tell from the block structure that there is a maximal block boundary somewhere between *Baringdorf* and *the*, it is

[3] I.e., an element of $\alpha(e_{i-1})$ (or $\alpha(e_{i-1})$) continues the L_2-string at the other end.

[4] We will come back to the situation where a block boundary may not be unique below.

[5] This will be explained below.

[6] The average sentence length for the English sentence is 26.5 words. (Not too suprisingly, Swedish gives rise to the fewest divergences against English. Note also that the Romance languages shown here behave very similarly.)

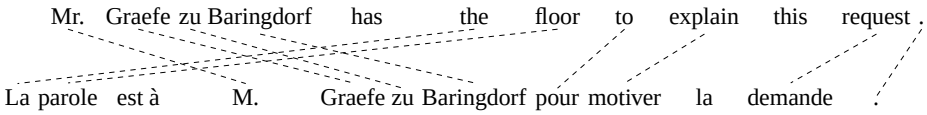


Figure 3: Alignment example with zero-fertility word in English

unclear on which side *has* should be located.[7]

The definitions of the various types of word blocks cover zero fertility words in principle, but they are somewhat awkward in that the same word may belong to two maximal α-blocks, on its left and on its right. It is not clear where the exact block boundary is located. So we redefine the notion of α-block boundaries. We call the (possibly empty) substring between the rightmost non-zero-fertility word of one maximal α-block and the leftmost non-zero-fertility word of its right neighbor block the **α-block boundary zone**.

The distituent hypothesis is sensitive to *crossing* a boundary zone, i.e., if a constituent-candidate ends somewhere in the middle of a non-empty boundary zone, this does not count as a crossing. This reflects the intuition of uncertainty and keeps the exclusion of clear distituents intact.

3 EM grammar induction with weighting factors

The distituent identification scheme introduced in the previous section can be used to hypothesize a fairly reliable exclusion of constituency for many spans of strings from a parallel corpus. Besides a statistical word alignment, no further resources are required.

In order to make use of this scattered (non-) constituency information, a semi-supervised approach is needed that can fill in the (potentially large) areas for which no prior information is available. For the present experiments we decided to choose a conceptually simple such approach, with which we can build on substantial existing work in grammar induction: we construe the learning problem as PCFG induction, using the inside-outside algorithm, with the addition of weighting factors based on the (non-)constituency information. This use of weighting factors in EM learning follows the approach discussed in (Nigam et al., 2000).

Since we are mainly interested in comparative experiments at this stage, the conceptual simplicity, and the availability of efficient implemented open-source systems of a PCFG induction approach outweighs the disadvantage of potentially poorer overall performance than one might expect from some other approaches.

The PCFG topology we use is a binary, entirely unrestricted X-bar-style grammar based on the Penn Treebank POS-tagset (expanded as in the TreeTagger by (Schmid, 1994)). All possible combinations of projections of POS-categories X and Y are included following the schemata in (5). This gives rise to 13,110 rules.

(5) a. $XP \rightarrow X$
b. $XP \rightarrow XP\ YP$
c. $XP \rightarrow YP\ XP$
d. $XP \rightarrow YP\ X$
e. $XP \rightarrow X\ YP$

We tagged the English version of our training section of the Europarl corpus with the TreeTagger and used the strings of POS-tags as the training corpus for the inside-outside algorithm; however, it is straightforward to apply our approach to a language for which no taggers are available if an unsupervised word clustering technique is applied first.

We based our EM training algorithm on Mark Johnson's implementation of the inside-outside algorithm.[8] The initial parameters on the PCFG rules are set to be uniform. In the iterative induction process of parameter reestimation, the current rule parameters are used to compute the expectations of how often each rule occurred in the parses of the training corpus, and these expectations are used to adjust the rule parameters, so that the likelihood of the training data is increased. When the probablity of a given rule drops below a certain threshold, the rule is excluded from the grammar. The iteration is continued until the increase in likelihood of the training corpus is very small.

Weight factors. The inside-outside algorithm is a dynamic programming algorithm that uses a chart in order to compute the rule expectations for each sentence. We use the information obtained from the parallel corpus as discussed in section 2 as prior information (in a Bayesian framework) to adjust the

[7] Since zero-fertility words are often function words, there is probably a rightward-tendency that one might be able to exploit; however in the present study we didn't want to build such high-level linguistic assumptions into the system.

[8] http://cog.brown.edu/~mj/

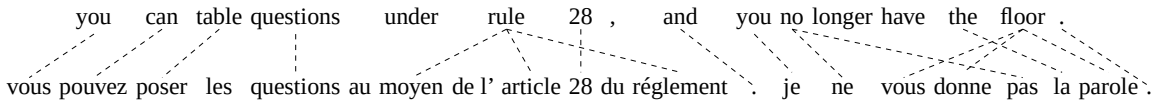


Figure 4: Alignment example with higher-fertility words in English

expectations that the inside-outside algorithm determines based on its current rule parameters. Note that the this prior information is information about string spans of (non-)constituents – it does not tell us anything about the categories of the potential constituents affected. It is combined with the PCFG expectations as the chart is constructed. For each span in the chart, we get a weight factor that is multiplied with the parameter-based expectations.[9]

4 Experiments

We applied GIZA++ (Al-Onaizan et al., 1999; Och and Ney, 2003) to word-align parts of the Europarl corpus (Koehn, 2002) for English and all other 10 languages. For the experiments we report in this paper, we only used the 1999 debates, with the language pairs of English combined with Finnish, French, German, Greek, Italian, Spanish, and Swedish.

For computing the weight factors we used a two-step process implemented in Perl, which first determines the maximal α-block boundaries (by detecting discontinuities in the sequence of the α-projected words). Words with fertility > 1 whose α-correspondents were non-adjacent (modulo NULL-projections) were treated like zero fertility words, i.e., we viewed them as unreliable indicators of block status (compare figure 4). (7) shows the internal representation of the block structure for (6) (compare figure 3). `L` and `R` are used for the beginning and end of blocks, when the adjacent boundary zone is empty; `l` and `r` are used next to non-empty boundary zones. Words that have correspondents in the normal sequence are encoded as `*`, zero fertility words as `-`; `A` and `B` are used for the first block in a sentence instead of `L` and `R`, unless it arises from "relocation", which increases likelihood for constituent status (likewise for the last block: `Y` and `Z`). Since we are interested only in first-order blocks here, the compact string-based representation is sufficient.

(6)
```
la parole est à m. graefe zu baring-
dorf pour motiver la demande
NULL ({ 3 4 11 }) mr ({ 5 }) graefe
({ 6 }) zu ({ 7 }) baringdorf ({ 8 })
has ({ }) the ({ 1 }) floor ({ 2 })
to ({ 9 }) explain ({ 10 }) this ({ })
request ({ 12 })
```

(7) `[L**r-lRY*-*Z]`

The second step for computing the weight factors creates a chart of all string spans over the given sentence and marks for each span whether it is a distituent, possible constituent or likely distituent, based on the location of boundary symbols. (For instance *zu Baringdorf has the* is marked as a distituent; *the floor* and *has the floor* are marked as likely constituents.) The tests are implemented as simple regular expressions. The chart of weight factors is represented as an array which is stored in the training corpus file along with the sentences. We combine the weight factors from various languages, since each of them may contribute distinct (non-)constituent information. The inside-outside algorithm reads in the weight factor array and uses it in the computation of expected rule counts.

We used the probability of the statistical word alignment as a confidence measure to filter out unreliable training sentences. Due to the conservative nature of the information we extract from the alignment, the results indicate however that filtering is not necessary.

5 Evaluation

For evaluation, we ran the PCFG resulting from training with the Viterbi algorithm[10] on parts of the Wall Street Journal (WSJ) section of the Penn Treebank and compared the tree structure for the most

[9] In the simplest model, we use the factor 0 for spans satisfying the distituent condition underlying hypothesis (4), and factor 1 for all other spans; in other words, parses involving a distituent are cancelled out. We also experimented with various levels of weight factors: for instance, distituents were assigned factor 0.01, likely distituents factor 0.1, neutral spans 1, and likely constituents factor 2. Likely constituents are defined as spans for which one end is adjacent to an *empty* block boundary zone (i.e., there is no zero fertility word in the block boundary zone which could be the actual boundary of constituents in which the block is involved).

Most variations in the weighting scheme did not have a significant effect, but they caused differences in coverage because rules with a probability below a certain threshold were dropped in training. Below, we report the results of the 0.01–0.1–1–2 scheme, which had a reasonably high coverage on the test data.

[10] We used the LoPar parser (Schmid, 2000) for this.

System	Unlab. Prec.	Unlab. Recall	F_1-Score	Crossing Brack.
Left-branching	30.4	35.8	32.9	3.06
Right-branching	36.2	42.6	39.2	2.48
Standard PCFG induction	42.4	64.9	51.3	2.2
PCFG trained with C/D weight factors from Europarl corpus	**47.8**	**72.1**	**57.5**	**1.7**
Upper limit	66.08	100.0	79.6	0.0

Figure 5: Scores for test sentences from WSJ section 23, up to length 10.

probable parse for the test sentences against the gold standard treebank annotation. (Note that one does not necessarily expect that an induced grammar will match a treebank annotation, but it may at least serve as a basis for comparison.) The evaluation criteria we apply are unlabeled bracketing precision and recall (and crossing brackets). We follow an evaluation criterion that (Klein and Manning, 2002, footnote 3) discuss for the evaluation of a not fully supervised grammar induction approach based on a binary grammar topology: bracket multiplicity (i.e., non-branching projections) is collapsed into a single set of brackets (since what is relevant is the constituent structure that was induced).[11] For comparison, we provide baseline results that a uniform left-branching structure and a uniform right-branching structure (which encodes some non-trivial information about English syntax) would give rise to. As an upper boundary for the performance a binary grammar can achieve on the WSJ, we present the scores for a minimal binarized extension of the gold-standard annotation.

The results we can report at this point are based on a comparatively small training set.[12] So, it may be too early for conclusive results. (An issue that arises with the small training set is that smoothing techniques would be required to avoid overtraining, but these tend to dominate the test application, so the effect of the parallel-corpus based information cannot be seen so clearly.) But we think that the results are rather encouraging.

As the table in figure 5 shows, the PCFG we induced based on the parallel-text derived weight factors reaches 57.5 as the F_1-score of unlabeled precision and recall on sentences up to length 10.[13] We show the scores for an experiment without smoothing, trained on c. 3,000 sentences. Since no smoothing was applied, the resulting coverage (with low-probability rules removed) on the test set is about 80%. It took 74 iterations of the inside-outside algorithm to train the weight-factor-trained grammar; the final version has 1005 rules.

For comparison we induced another PCFG based on the same X-bar topology without using the weight factor mechanism. This grammar ended up with 1145 rules after 115 iterations. The F_1-score is only 51.3 (while the coverage is the same as for the weight-factor-trained grammar).

Figure 6 shows the complete set of (singular) "NP rules" emerging from the weight-factor-trained grammar, which are remarkably well-behaved, in particular when we compare them to the corresponding rules from the PCFG induced in the standard way (figure 7). (XP categories are written as ⟨POS-TAG⟩-P, X head categories are written as ⟨POS-TAG⟩-0 – so the most probable NP productions in figure 6 are NP → N PP, NP → N, NP → ADJP N, NP → NP PP, NP → N PropNP.)

Of course we are comparing an unsupervised technique with a mildly supervised technique; but the results indicate that the relatively subtle information discussed in section 2 seems to be indeed very useful.

6 Discussion

This paper presented a novel approach of using parallel corpora as the only resource in the creation of a monolingual analysis tools. We believe that in order to induce high-quality tools based on statistical word alignment, the training approach for the target language tool has to be able to exploit islands of reliable information in a stream of potentially rather noisy data. We experimented with an initial idea to address this task, which is conceptually simple and can be implemented building on existing technology: using the notion of word blocks projected

[11] Note that we removed null elements from the WSJ, but we left punctuation in place. We used the EVALB program for obtaining the measures, however we preprocessed the bracketings to reflect the criteria we discuss here.

[12] This is not due to scalability issues of the system; we expect to be able to run experiments on rather large training sets. Since no manual annotation is required, the available resources are practically indefinite.

[13] For sentences up to length 30, the F_1-score drops to 28.7 (as compared to 23.5 for the standard PCFG).

```
0.300467        NN-P --> NN-0 IN-P
0.25727         NN-P --> NN-0
0.222335        NN-P --> JJ-P NN-0
0.0612312       NN-P --> NN-P IN-P
0.0462079       NN-P --> NN-0 NP-P
0.0216048       NN-P --> NN-0 ,-P
0.0173518       NN-P --> NN-P NN-0
0.0114746       NN-P --> NN-0 NNS-P
0.00975112      NN-P --> NN-0 MD-P
0.00719605      NN-P --> NN-0 VBZ-P
0.00556762      NN-P --> NN-0 NN-P
0.00511326      NN-P --> NN-0 VVD-P
0.00438077      NN-P --> NN-P VBD-P
0.00423814      NN-P --> NN-P ,-P
0.00409675      NN-P --> NN-0 CD-P
0.00286634      NN-P --> NN-0 VHZ-P
0.00258022      NN-P --> VVG-P NN-0
0.0018237       NN-P --> NN-0 TO-P
0.00162601      NN-P --> NN-P VVN-P
0.00157752      NN-P --> NN-P VB-P
0.00125101      NN-P --> NN-0 VVN-P
0.00106749      NN-P --> NN-P VBZ-P
0.00105866      NN-P --> NN-0 VBD-P
0.000975359     NN-P --> VVN-P NN-0
0.000957702     NN-P --> NN-0 SENT-P
0.000931056     NN-P --> NN-0 CC-P
0.000902116     NN-P --> NN-P SENT-P
0.000717542     NN-P --> NN-0 VBP-P
0.000620843     NN-P --> RB-P NN-0
0.00059608      NN-P --> NN-0 WP-P
0.000550255     NN-P --> NN-0 PDT-P
0.000539155     NN-P --> NN-P CC-P
0.000341498     NN-P --> WP$-P NN-0
0.000330967     NN-P --> WRB-P NN-0
0.000186441     NN-P --> ,-P NN-0
0.000135449     NN-P --> CD-P NN-0
7.16819e-05     NN-P --> NN-0 POS-P
```

Figure 6: Full set of rules based on the NN tag in the C/D-trained PCFG

```
0.429157        NN-P --> DT-P NN-0
0.0816385       NN-P --> IN-P NN-0
0.0630426       NN-P --> NN-0
0.0489261       NN-P --> PP$-P NN-0
0.0487434       NN-P --> JJ-P NN-0
0.0451819       NN-P --> NN-P ,-P
0.0389741       NN-P --> NN-P VBZ-P
0.0330732       NN-P --> NN-P NN-0
0.0215872       NN-P --> NN-P MD-P
0.0201612       NN-P --> NN-P TO-P
0.0199536       NN-P --> CC-P NN-0
0.015509        NN-P --> NN-P VVZ-P
0.0112734       NN-P --> NN-P RB-P
0.00977683      NN-P --> NP-P NN-0
0.00943218      NN-P --> CD-P NN-0
0.00922132      NN-P --> NN-P WDT-P
0.00896826      NN-P --> POS-P NN-0
0.00749452      NN-P --> NN-P VHZ-P
0.00621328      NN-P --> NN-0 ,-P
0.00520734      NN-P --> NN-P VBD-P
0.004674        NN-P --> JJR-P NN-0
0.00407644      NN-P --> NN-P VVD-P
0.00394681      NN-P --> NN-P VVN-P
0.00354741      NN-P --> NN-0 MD-P
0.00335451      NN-P --> NN-0 NN-P
0.0030748       NN-P --> EX-P NN-0
0.0026483       NN-P --> WRB-P NN-0
0.00262025      NN-P --> NN-0 TO-P
[...]
0.000403279     NN-P --> NN-0 VBP-P
0.000378414     NN-P --> NN-0 PDT-P
0.000318026     NN-P --> NN-0 VHZ-P
2.27821e-05     NN-P --> NN-P PP-P
```

Figure 7: Standard induced PCFG: Excerpt of rules based on the NN tag

by word alignment as an indication for (mainly) impossible string spans. Applying this information in order to impose weighting factors on the EM algorithm for PCFG induction gives us a first, simple instance of the "island-exploiting" system we think is needed. More sophisticated models may make use some of the experience gathered in these experiments.

The conservative way in which cross-linguistic relations between phrase structure is exploited has the advantage that we don't have to make unwarranted assumptions about direct correspondences among the majority of constituent spans, or even direct correspondences of phrasal categories. The technique is particularly well-suited for the exploitation of parallel corpora involving multiple languages like the Europarl corpus. Note that nothing in our methodology made any language particular assumptions; future research has to show whether there are language pairs that are particularly effective, but in general the technique should be applicable for whatever parallel corpus is at hand.

A number of studies are related to the work we presented, most specifically work on parallel-text based "information projection" for parsing (Hwa et al., 2002), but also grammar induction work based on constituent/distituent information (Klein and Manning, 2002) and (language-internal) alignment-based learning (van Zaanen, 2000). However to our knowledge the specific way of bringing these aspects together is new.

References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical machine translation. Final report, JHU Workshop.

Michael Collins. 1999. A statistical parser for Czech. In *Proceedings of ACL*.

Rebecca Hwa, Philip Resnik, and Amy Weinberg. 2002. Breaking the resource bottleneck for multilingual parsing. In *Proceedings of LREC*.

Dan Klein and Christopher Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of ACL*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference 2003 (HLT-NAACL 2003)*, Edmonton, Canada.

Philipp Koehn. 2002. Europarl: A multilingual corpus for evaluation of machine translation. Ms., University of Southern California.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK.

Helmut Schmid. 2000. Lopar: Design and implementation. Arbeitspapiere des Sonderforschungsbereiches 340, No. 149, IMS Stuttgart.

Menno van Zaanen. 2000. ABL: Alignment-based learning. In *COLING 2000 - Proceedings of the 18th International Conference on Computational Linguistics*, pages 961–967.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

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