Parsing Word-Aligned Parallel Corpora in a Grammar Induction Context

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

Abstract

We present an Earley-style dynamic programming algorithm for parsing sentence pairs from a parallel corpus simultaneously, building up two phrase structure trees and a correspondence mapping between the nodes. The intended use of the algorithm is in bootstrapping grammars for less studied languages by using implicit grammatical information in parallel corpora. Therefore, we presuppose a given (statistical) word alignment underlying in the synchronous parsing task; this leads to a significant reduction of the parsing complexity. The theoretical complexity results are corroborated by a quantitative evaluation in which we ran an implementation of the algorithm on a suite of test sentences from the Europarl parallel corpus.

1 Introduction

The technical results presented in this paper[1] are motivated by the following considerations: It is conceivable to use sentence pairs from a parallel corpus (along with the tentative word correspondences from a statistical word alignment) as training data for a grammar induction approach. The goal is to induce monolingual grammars for the languages under consideration; but the implicit information about syntactic structure gathered from typical patterns in the alignment goes beyond what can be obtained from unlabeled monolingual data. Consider for instance the sentence pair from the Europarl corpus (Koehn, 2002) in fig. 1 (shown with a hand-labeled word alignment): distributional patterns over this and similar sentences may show that in English, the subject (the word block "*the situation*") is in a fixed structural position, whereas in German, it can appear in various positions; similarly, the finite verb in German (here: *stellt*) systematically appears in second position in main clauses. In a way, the translation of sentences into other natural languages serves as an approximation of a (much more costly) manual structural or semantic annotation – one might speak of automatic indirect supervision in learning. The technique will be most useful for low-resource languages and languages for which there is no funding for treebanking activities. The only requirement will be that a parallel corpus exist for the language under consideration and one or more other languages.[2]

Induction of grammars from parallel corpora is rarely viewed as a promising task in its own right; in work that has addressed the issue directly (Wu, 1997; Melamed, 2003; Melamed, 2004), the synchronous grammar is mainly viewed as instrumental in the process of improving the translation model in a noisy channel approach to statistical MT.[3] In the present paper, we provide an important prerequisite for parallel corpus-based grammar induction work: an efficient algorithm for synchronous parsing of sentence pairs, given a word alignment. This work represents a second pilot study (after (Kuhn, 2004)) for the longer-term PTOLEMAIOS project at Saarland University[4] with the goal of learning linguistic grammars from parallel corpora (compare (Kuhn, 2005)). The grammars should be robust and assign a

[1] This work was in part supported by the German Research Foundation DFG in the context of the author's Emmy Noether research group at Saarland University.

[2] In the present paper we use examples from English/German for illustration, but the approach is of course independent of the language pair under consideration.

[3] Of course, there is related work (e.g., (Hwa et al., 2002; Lü et al., 2002)) using aligned parallel corpora in order to "project" bracketings or dependency structures from English to another language and exploit them for training a parser for the other language. But note the conceptual difference: the "parse projection" approach departs from a given monolingual parser, with a particular style of analysis, whereas our project will explore to what extent it may help to design the grammar topology specifically for the parallel corpus case. This means that the emerging English parser may be different from all existing ones.

[4] http://www.coli.uni-saarland.de/˜jonask/PTOLEMAIOS/

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 17–24,
Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

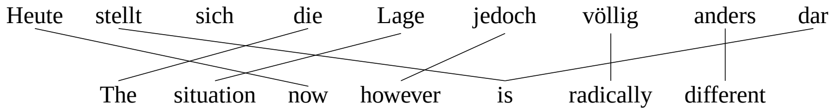

Figure 1: Word-aligned German/English sentence pair from the Europarl corpus

predicate-argument-modifier (or dependency) structure to sentences, such that they can be applied in the context of multilingual information extraction or question answering.

2 Synchronous grammars

For the purpose of grammar induction from parallel corpora, we assume a fairly straightforward extension of context-free grammars to the synchronous grammar case (compare the *transduction grammars* of (Lewis II and Stearns, 1968)): Firstly, the terminal and non-terminal categories are pairs of symbols, one for each language; as a special case, one of the two symbols can be NIL for material realized in only one of the languages. Secondly, the linear sequence of daughter categories that is specified in the rules can differ for the two languages; therefore, an explicit numerical ranking is used for the linear precedence in each language. We use a compact rule notation with a numerical ranking for the linear precedence in each language. The general form of a grammar rule for the case of two parallel languages is $N_0/M_0 \rightarrow N_1{:}i_1/M_1{:}j_1 \ldots N_k{:}i_k/M_k{:}j_k$, where N_l, M_l are NIL or a terminal or nonterminal symbol for language L_1 and L_2, respectively, and i_l, j_l are natural numbers for the rank of the phrase in the sequence for L_1 and L_2 respectively (for NIL categories a special rank 0 is assumed).[5] Since linear ordering of daughters in both languages is explicitly encoded by the rank indices, the specification sequence in the rule is irrelevant from a declarative point of view. To facilitate parsing we assume a normal form in which the right-hand side is ordered by the rank in L_1, with the exception that the categories that are NIL in L_1 come last. If there are several such NIL categories in the same rule, they are viewed as unordered with respect to each other.[6]

Fig. 2 illustrates our simple synchronous grammar formalism with some rules of a sample grammar and their application on a German/English sentence pair. Derivation with a synchronous grammar gives rise to a multitree, which combines classical phrase structure trees for the languages involved and also encodes the phrase level correspondence across the languages. Note that the two monolingual trees in fig. 2 for German and English are just two ways of unfolding the common underlying multitree.

Note that the simple formalism goes along with the **continuity assumption** that *every complete constituent is continuous in both languages*. Various recent studies in the field of syntax-based Statistical MT have shown that such an assumption is problematic when based on typical treebank-style analyses. As (Melamed, 2003) discusses for instance, in the context of binary branching structures even simple examples like the English/French pair *a gift for you from France* ↔ *un cadeau de France pour vouz* [*a gift from France for you*] lead to discontinuity of a "synchronous phrase" in one of the two languages. (Gildea, 2003) and (Galley et al., 2004) discuss different ways of generalizing the tree-level crosslinguistic correspondence relation, so it is not confined to single tree nodes, thereby avoiding a continuity assumption. We believe that in order to obtain full coverage on real parallel corpora, some mechanism along these lines will be required.

However, if the typical rich phrase structure analyses (with fairly detailed fine structure) are replaced by flat, multiply branching analyses, most of the highly frequent problematic cases are resolved.[7] In

[5] Note that in the probabilistic variants of these grammars, we will typically expect that *any* ordering of the right-hand side symbols is possible (but that the probability will of course vary – in a maximum entropy or log-linear model, the probability will be estimated based on a variety of learning features). This means that in parsing, the right-hand side categories will be accepted as they come in, and the relevant probability parameters are looked up accordingly.

[6] This detail will be relevant for the parsing inference rule (5) below.

[7] Compare the systematic study for English-French alignments by (Fox, 2002), who compared (i) treebank-parser style analyses, (ii) a variant with flattened VPs, and (iii) dependency structures. The degree of cross-linguistic phrasal cohesion increases from (i) to (iii). With flat clausal trees, we will come close to dependency structures with respect to cohesion.

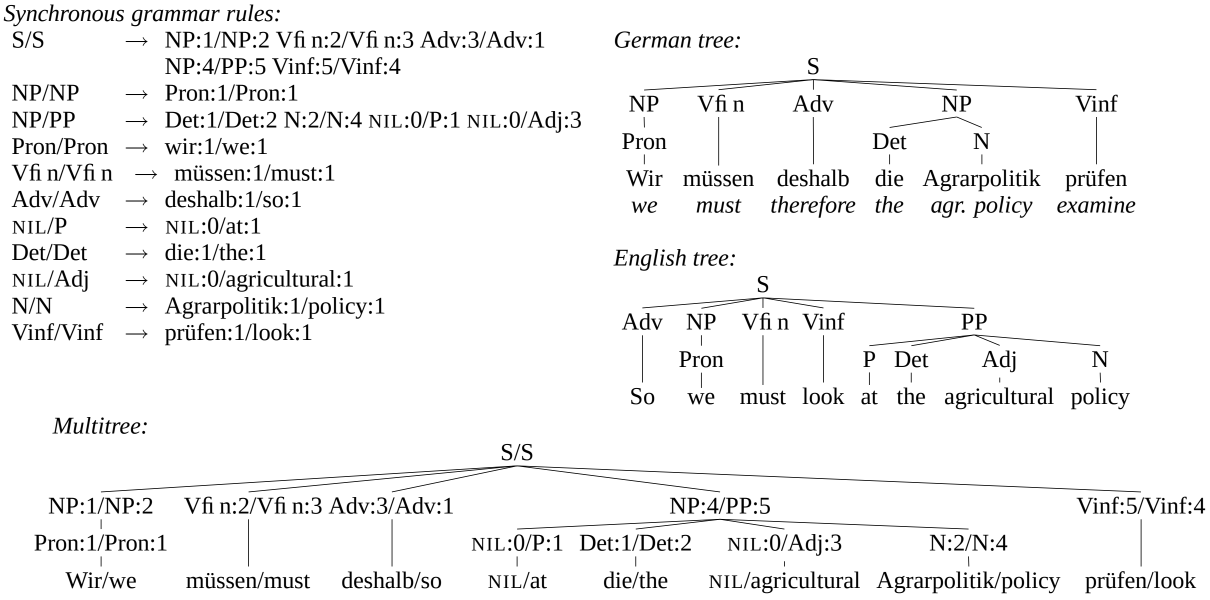

Figure 2: Sample rules and analysis for a synchronous grammar

the flat representation that we assume, a clause is represented in a single subtree of depth 1, with all verbal elements and the argument/adjunct phrases (NPs or PPs) as immediate daughters of the clause node. Similarly, argument/adjunct phrases are flat internally. Such a flat representation is justified both from the point of view of linguistic learning and from the point of view of grammar application: (i) Language-specific principles of syntactic structure (e.g., the strong configurationality of English), which are normally captured linguistically by the richer phrase structure, are available to be induced in learning as systematic patterns in the relative ordering of the elements of a clause. (ii) The predicate-argument-modifier structure relevant for application of the grammars, e.g., in information extraction can be directly read off the flat clausal representation.

It is a hypothesis of our longer-term project that a word alignment-based consensus structure which works with flat representations and under the continuity assumption is a very effective starting point for learning the basic language-specific constraints required for a syntactic grammar. Linguistic phenomena that fall outside what can be captured in this confined framework (in particular unbounded dependencies spanning more than one clause and discontinuous argument phrases) will then be learned in a later bootstrapping step that provides a richer set of operations. We are aware of a number of open practical questions, e.g.: Will the fact that real parallel corpora often contain rather free translations undermine our idea of using the consensus structure for learning basic syntactic constraints? Statistical alignments are imperfect – can the constraints imposed by the word alignment be relaxed accordingly without sacrificing tractability and the effect of indirect supervision?[8]

3 Alignment-guided synchronous parsing

Our dynamic programming algorithm can be described as a variant of standard Earley-style chart parsing (Earley, 1970) and generation (Shieber, 1988; Kay, 1996). The chart is a data structure which stores all sub-analyses that cover part of the input string (in parsing) or meaning representation (in generation). Memoizing such partial results has the standard advantage of dynamic programming techniques – it helps one to avoid unnecessary recomputation of partial results. The chart structure for context-free parsing is also exploited directly in dynamic programming algorithms for probabilistic context-free grammars (PCFGs): (i) the inside (or outside) algorithm for summing over the probabilities for every possible analysis of a given string, (ii) the Viterbi algorithm for determining the most likely analysis of a given string, and (iii) the in-

[8] Ultimately, bootstrapping of not only the grammars, but also of the word alignment should be applied.

side/outside algorithm for re-estimating the parameters of the PCFG in an Expectation-Maximization approach (i.e., for iterative training of a PCFG on unlabeled data). This aspect is important for the intended later application of our parsing algorithm in a grammar induction context.

A convenient way of describing Earley-style parsing is by inference rules. For instance, the central *completion* step in Earley parsing can be described by the rule[9]

(1) $$\frac{\langle X \rightarrow \alpha \bullet Y\,\beta, [i,j]\rangle,\ \langle Y \rightarrow \gamma\,\bullet, [j,k]\rangle}{\langle X \rightarrow \alpha\, Y \bullet \beta, [i,k]\rangle}$$

Synchronous parsing. The input in synchronous parsing is not a one-dimensional string, but a pair of sentences, i.e., a two-dimensional array of possible word pairs (or a multidimensional array if we are looking at a multilingual corpus), as illustrated in fig. 3.

policy					●	
agricultural						
the				●		
at						
look						●
must		●				
we	●					
So			●			
	0	1	2	3	4	5 6
L2: L1:	Wir	müssen	deshalb	die	Agrar-politik	prüfen

Figure 3: Synchronous parsing: two-dimensional input (with word alignment marked)

The natural way of generalizing context-free parsing to synchronous grammars is thus to control the inference rules by string indices in both dimensions. Graphically speaking, parsing amounts to identifying rectangular crosslinguistic constituents – by assembling smaller rectangles that will together cover the full string spans in both dimensions (compare (Wu, 1997; Melamed, 2003)). For instance in fig. 4, the NP/NP rectangle $[i_1, j_1, j_2, k_2]$ can be combined with the Vinf/Vinf rectangle $[j_1, k_1, i_2, j_2]$ (assuming there is an appropriate rule in the grammar).

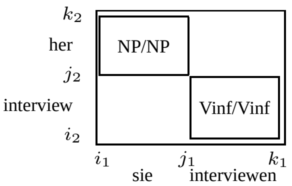

Figure 4: Completion in two-dimensional chart: parsing part of *Can I interview her?/Kann ich sie interviewen?*

More generally, we get the inference rules (2) and (3) (one for the case of parallel sequencing, one for crossed order across languages).

(2) $$\frac{\begin{array}{c}\langle X_1/X_2 \rightarrow \alpha \bullet Y_1{:}r_1/Y_2{:}r_2\ \beta, [i_1, j_1, i_2, j_2]\rangle,\\ \langle Y_1/Y_2 \rightarrow \gamma\,\bullet, [j_1, k_1, j_2, k_2]\rangle\end{array}}{\langle X_1/X_2 \rightarrow \alpha\, Y_1{:}r_1/Y_2{:}r_2 \bullet \beta, [i_1, k_1, i_2, k_2]\rangle}$$

(3) $$\frac{\begin{array}{c}\langle X_1/X_2 \rightarrow \alpha \bullet Y_1{:}r_1/Y_2{:}r_2\ \beta, [i_1, j_1, j_2, k_2]\rangle,\\ \langle Y_1/Y_2 \rightarrow \gamma\,\bullet, [j_1, k_1, i_2, j_2]\rangle\end{array}}{\langle X_1/X_2 \rightarrow \alpha\, Y_1{:}r_1/Y_2{:}r_2 \bullet \beta, [i_1, k_1, i_2, k_2]\rangle}$$

Since each inference rule contains six free variables over string positions ($i_1, j_1, k_1, i_2, j_2, k_2$), we get a parsing complexity of order $O(n^6)$ for unlexicalized grammars (where n is the number of words in the longer of the two strings from language L_1 and L_2) (Wu, 1997; Melamed, 2003). For large-scale learning experiments this may be problematic, especially when one moves to lexicalized grammars, which involve an additional factor of n^4.[10]

As a further issue, we observe that the inference rules are insufficient for multiply branching rules, in which *partial constituents* may be discontinuous in one dimension (only complete constituents need to be continuous in both dimensions). For instance, by parsing the first two words of the German string in fig. 1 (*Heute stellt*), we should get a partial chart item for a sentence, but the English correspondents for the two words (*now* and *is*) are discontinuous, so we couldn't apply rule (2) or (3).

Correspondence-guided parsing. As an alternative to the standard "rectangular indexing" approach

[9] A chart item is specified through a position (•) in a production and a string span ($[l_1, l_2]$). $\langle X \rightarrow \alpha \bullet Y\beta, [i,j]\rangle$ means that between string position i and j, the beginning of an X phrase has been found, covering α, but still missing $Y\beta$. Chart items for which the dot is at the end of a production (like $\langle Y \rightarrow \gamma\bullet, [j,k]\rangle$) are called passive items, the others active.

[10] The assumption here (following (Melamed, 2003)) is that lexicalization is not considered as just affecting the grammar constant, but that in parsing, every terminal symbol has to be considered as the potential head of every phrase of which it is a part. Melamed demonstrates: If the number of different category symbols is taken into consideration as l, we get $O(l^2n^6)$ for unlexicalized grammars, and $O(l^6n^{10})$ for lexicalized grammars; however there are some possible optimizations.

to synchronous parsing we propose a conceptually very simple asymmetric approach. As we will show in sec. 4 and 5, this algorithm is both theoretically and practically efficient when applied to sentence pairs for which a word alignment has previously been determined. The approach is asymmetric in that one of the languages is viewed as the "master language", i.e., indexing in parsing is mainly based on this language (the "primary index" is the string span in L_1 as in monolingual parsing). The other language contributes a secondary index, which is mainly used to guide parsing in the master language – i.e., certain options are eliminated. The choice of the master language is in principle arbitrary, but for efficiency considerations it is better to pick the one that has more words without a correspondent.

A way of visualizing correspondence-guided parsing is that standard Earley *parsing* is applied to L_1, with primary indexing by string position; as the chart items are assembled, the synchronous grammar and the information from the word alignment is used to check whether the string in L_2 could be generated (essentially using chart-based *generation* techniques; cf. (Shieber, 1988; Neumann, 1998)). The index for chart items consists of two components: the string span in L_1 and a bit vector for the words in L_2 which are covered. For instance, based on fig. 3, the noun compound *Agrarpolitik* corresponding to *agricultural policy* in English will have the index $\langle[4,5],[0,0,0,0,0,0,1,1]\rangle$ (assuming for illustrative purposes that German is the master language in this case).

The completion step in correspondence-guided parsing can be formulated as the following single inference rule:[11]

(4) $$\frac{\langle X_1/X_2 \rightarrow \alpha \bullet Y_1{:}r_1/Y_2{:}r_2\ \ \beta, \langle[i,j],\mathbf{v}\rangle\rangle, \quad \langle Y_1/Y_2 \rightarrow \gamma \bullet, \langle[j,k],\mathbf{w}\rangle\rangle}{\langle X_1/X_2 \rightarrow \alpha\ Y_1{:}r_1/Y_2{:}r_2 \bullet \beta, \langle[i,k],\mathbf{u}\rangle\rangle}$$

where

(i) $j \neq k$;
(ii) $\text{OR}(\mathbf{v},\mathbf{w}) = \mathbf{u}$;
(iii) $\mathbf{w}$ is continuous (i.e., it contains maximally one subsequence of 1's).

Condition (iii) excludes discontinuity in passive chart items, i.e., complete constituents; active items (i.e., partial constituents) may well contain discontinuities. The success condition for parsing a string with N words in L_1 is that a chart item with index $\langle[0,N],\mathbf{1}\rangle$ has been found for the start category pair of the grammar.

Words in L_2 with no correspondent in L_1 (let's call them "L_1-NIL"s for short), for example the words *at* and *agricultural* in fig. 3,[12] can in principle appear between any two words of L_1. Therefore they are represented with a "variable" empty L_1-string span like for instance in $\langle[i,i],[0,0,1,0,0]\rangle$. At first blush, such L_1-NILs seem to introduce an extreme amount of non-determinism into the algorithm. Note however that due to the continuity assumption for complete constituents, the distribution of the L_1-NILs is constrained by the other words in L_2. This is exploited by the following inference rule, which is the only way of integrating L_1-NILs into the chart:

(5) $$\frac{\langle X_1/X_2 \rightarrow \alpha \bullet \text{NIL}{:}0/Y_2{:}r_2\ \ \beta, \langle[i,j],\mathbf{v}\rangle\rangle, \quad \langle \text{NIL}/Y_2 \rightarrow \gamma \bullet, \langle[j,j],\mathbf{w}\rangle\rangle}{\langle X_1/X_2 \rightarrow \alpha\ \text{NIL}{:}0/Y_2{:}r_2 \bullet \beta, \langle[i,j],\mathbf{u}\rangle\rangle}$$

where

(i) $\mathbf{w}$ is adjacent to $\mathbf{v}$ (i.e., unioning vectors $\mathbf{w}$ and $\mathbf{v}$ does not lead to more 0-separated 1-sequences than $\mathbf{v}$ contains already);
(ii) $\text{OR}(\mathbf{v},\mathbf{w}) = \mathbf{u}$.

The rule has the effect of finalizing a cross-linguistic constituent (i.e., rectangle in the two-dimensional array) after all the parts that have correspondents in both languages have been found. [13]

4 Complexity

We assume that the two-dimensional chart is initialized with the correspondences following from a word alignment. Hence, for each terminal that is non-empty in L_1, both components of the index are known. When two items with known secondary indices are combined with rule (4), the new secondary

[11] We use the bold-faced variables $\mathbf{v},\mathbf{w},\mathbf{u}$ for bit vectors; the function OR performs bitwise disjunction on the vectors (e.g., $\text{OR}([0,1,1,0,0],[0,0,1,0,1]) = [0,1,1,0,1]$).

[12] It is conceivable that a word alignment would list *agricultural* as an additional correspondent for *Agrarpolitik*; but we use the given alignment for illustrative purposes.

[13] For instance, the L_1-NILs in fig. 3 – NIL/*at* and NIL/*agricultural* – have to be added to incomplete NP/PP constituent in the L_1-string span from 3 to 5, consisting of the Det/Det *die/the* and the N/N *Agrarpolitik/policy*. With two applications of rule (5), the two L_1-NILs can be added. Note that the conditions are met, and that as a result, we will have a continuous NP/PP constituent with index $\langle[3,5],[0,0,0,0,1,1,1,1]\rangle$, which can be used as a passive item Y_1/Y_2 in rule (4).

index can be determined by bitwise disjunction of the bit vectors. This operation is linear in the length of the L_2-string (which is of the same order as the length of the L_1-string) and has a very small constant factor.[14] Since parsing with a simple, non-lexicalized context-free grammar has a time complexity of $O(n^3)$ (due to the three free variables for string positions in the completion rule), we get $O(n^4)$ for *synchronous parsing of sentence pairs without any L_1-NILs*. Note that words from L_1 without a correspondent in L_2 (which we would have to call L_2-NILs) do not add to the complexity, so the language with more correspondent-less words can be selected as L_1.

For the *average complexity* of correspondence-guided parsing of sentence pairs without L_1-NILs we note an advantage over monolingual parsing: certain hypotheses for complete constituents that would have to be considered when parsing only L_1, are excluded because the secondary index reveals a discontinuity. An example from fig. 3 would be the sequence *müssen deshalb*, which is adjacent in L_1, but doesn't go through as a continuous rectangle when L_2 is taken into consideration (hence it cannot be used as a passive item in rule (4)).

The complexity of correspondence-guided parsing is certainly increased by the presence of L_1-NILs, since with them the secondary index can no longer be uniquely determined. However, with the adjacency condition ((i) in rule (5)), the number of possible variants in the secondary index is a function of the number of L_1-NILs. Let us say there are m L_1-NILs, i.e., the bit vectors contain m elements that we have to flip from 0 to 1 to obtain the final bit vector. In each application of rule (5) we pick a vector **v**, with a variable for the leftmost and rightmost L_1-NIL element (since this is not fully determined by the primary index). By the adjacency condition, either the leftmost or rightmost marks the boundary for adding the additional L_1-NIL element NIL/Y_2 – hence we need only one new variable for the newly shifted boundary among the L_1-NILs. So, in addition to the n^4 expense of parsing non-nil words, we get an expense of m^3 for parsing the L_1-NILs, and we conclude that for unlexicalized synchronous parsing, guided by an initial word alignment the complexity class is $O(n^4m^3)$ (where n is the total number of words appearing in L_1, and m is the number of words appearing in L_2, without a correspondent in L_1). Recall that the complexity for standard synchronous parsing is $O(n^6)$.

Since typically the number of correspondent-less words is significantly lower than the total number of words (at least for one of the two languages), these results are encouraging for medium-to-large-scale grammar learning experiments using a synchronous parsing algorithm.

5 Empirical Evaluation

In order to validate the theoretical complexity results empirically, we implemented the algorithm and ran it on sentence pairs from the Europarl parallel corpus. At the present stage, we are interested in quantitative results on parsing time, rather than qualitative results of parsing accuracy (for which a more extensive training of the rule parameters would be required).

Implementation. We did a prototype implementation of the correspondence-guided parsing algorithm in SWI Prolog.[15] Chart items are asserted to the knowledge base and efficiently retrieved using indexing by a hash function. Besides chart construction, the Viterbi algorithm for selecting the most probable analysis has been implemented, but for the current quantitative results only chart construction was relevant.

Sample grammar extraction. The initial probablistic grammar for our experiments was extracted from a small "multitree bank" of 140 German/English sentence pairs (short examples from the Europarl corpus). The multitree bank was annotated using the MMAX2 tool[16] and a specially

[14] Note that the operation does not have to be repeated when the completion rule is applied on additional pairs of items with identical indices. This means that the extra time complexity factor of n doesn't go along with an additional factor of the grammar constant (which we are otherwise ignoring in the present considerations). In practical terms this means that changes in the size of the grammar are much more noticable than moving from monolingual parsing to alignment-guided parsing.

An additional advantage is that in an Expectation Maximization approach to grammar induction (with a fixed word alignment), the bit vectors have to be computed only in the first iteration of parsing the training corpus, later iterations are cubic.

[15] http://www.swi-prolog.org – The advantage of using Prolog is that it is very easy to experiment with various conditions on the inference rules in parsing.

[16] http://mmax.eml-research.de

tailored annotation scheme for flat correspondence structures as described in sec. 2. A German and English part-of-speech tagger was used to determine word categories; they were mapped to a reduced category set and projected to the syntactic constituents.

To obtain parameters for a probabilistic grammar, we used maximum likelihood estimation from the small corpus, based on a rather simplistic generative model,[17] which for each local subtree decides (i) what categories will be the two heads, (ii) how many daughters there will be, and for each non-head sister (iii) whether it will be a nonterminal or a terminal (and in that case, what category pair), and (iv) in which position relative to the head to place it in both languages. In order to obtain a realistically-sized grammar, we applied smoothing to all parameters; so effectively, every sequence of terminals/nonterminals of arbitrary length was possible in parsing.

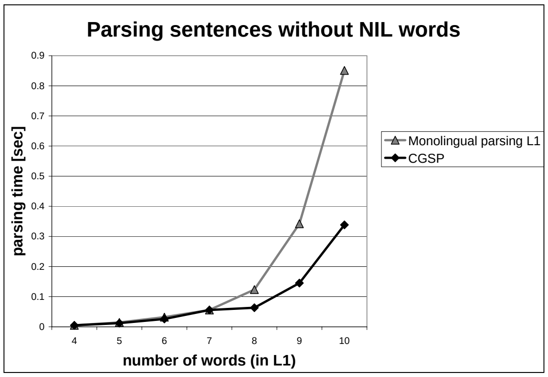

Figure 5: Comparison of synchronous parsing with and without exploiting constraints from L_2

Results. To validate empirically that the proposed correspondence-guided synchronous parsing approach (CGSP) can effectively exploit L_2 as a guide, thereby reducing the search space of L_1 parses that have to be considered, we first ran a comparison on sentences without L_1-NILs. The results (average parsing time for Viterbi parsing with the sample grammar) are shown in fig. 5.[18] The parser we call "monolingual" cannot exploit any alignment-induced restrictions from L_2.[19] Note that CGSP takes clearly less time.

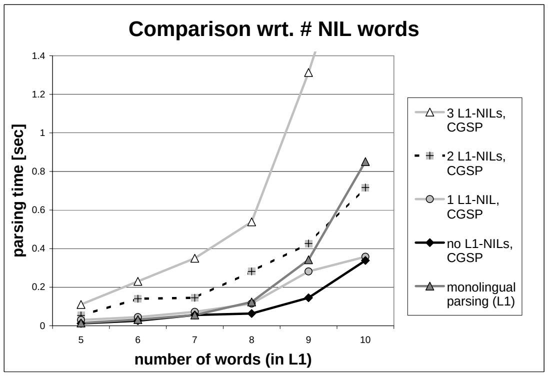

Figure 6: Synchronous parsing with a growing number of L_1-NILs

Fig. 6 shows our comparative results for parsing performance on sentences that do contain L_1-NILs. Here too, the theoretical results are corroborated that with a limited number of L_1-NILs, the CGSP is still efficient.

The average chart size (in terms of the number of entries) for sentences of length 8 (in L_1) was 212 for CGSP (and 80 for "monolingual" parsing). The following comparison shows the effect of L_1-NILs (note that the values for 4 and more L_1-NILs are based on only one or two cases):

(6) *Chart size for sentences of length 8 (in L_1)*

Number of L_1-NILs	0	1	2	3	4	5	6
Avg. number of chart items	77	121	175	256	(330)	(435)	(849)

We also simulated a synchronous parser which does not take advantage of a given word alignment (by providing an alignment link between any pair of words, plus the option that any word could be a NULL word). For sentences of length 5, this parser took an average time of 22.3 seconds (largely independent of the presence/absence of L_1-NILs).[20]

[17] For our learning experiments we intend to use a Maximum Entropy/log-linear model with more features.

[18] The experiments were run on a 1.4GHz Pentium M processor.

[19] The "monolingual" parser used in this comparison parses two identical copies of the same string synchronously, with a strictly linear alignment.

[20] While our simulation may be significantly slower than a direct implementation of the algorithm (especially when some of the optimizations discussed in (Melamed, 2003) are taken into account), the fact that it is orders of magnitude slower does in-

Finally, we also ran an experiment in which the continuity condition (condition (iii) in rule (4)) was deactivated, i.e., complete constituents were allowed to be discontinuous in one of the languages. The results in (7) underscore the importance of this condition – leaving it out leads to a tremendous increase in parsing time.

(7) *Average parsing time in seconds with and without continuity condition*

Sentence length (with no L_1-NILs)	4	5	6
Avg. parsing time with CGSP (incl. continuity condition)	0.005	0.012	0.026
Avg. parsing time without the continuity condition	0.035	0.178	1.025

6 Conclusion

We proposed a conceptually simple, yet efficient algorithm for synchronous parsing in a context where a word alignment can be assumed as given – for instance in a bootstrapping learning scenario. One of the two languages in synchronous parsing acts as the master language, providing the primary string span index, which is used as in classical Earley parsing. The second language contributes a bit vector as a secondary index, inspired by work on chart generation. Continuity assumptions make it possible to constrain the search space significantly, to the point that synchronous parsing for sentence pairs with few "NULL words" (which lack correspondents) may be faster than standard monolingual parsing. We discussed the complexity both theoretically and provided a quantitative evaluation based on a prototype implementation.

The study we presented is part of the longer-term PTOLEMAIOS project. The next step is to apply the synchronous parsing algorithm with probabilistic synchronous grammars in grammar induction experiments on parallel corpora.

References

Jay C. Earley. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 304–311.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 273–280.

Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03), Sapporo, Japan*, pages 80–87.

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

Martin Kay. 1996. Chart generation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA*.

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

Jonas Kuhn. 2004. Experiments in parallel-text based grammar induction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics: ACL 2004*, pages 470–477.

Jonas Kuhn. 2005. An architecture for parallel corpus-based grammar learning. In Bernhard Fisseni, Hans-Christian Schmitz, Bernhard Schröder, and Petra Wagner, editors, *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen. Beiträge zur GLDV-Tagung 2005 in Bonn*, pages 132–144, Frankfurt am Main. Peter Lang.

Philip M. Lewis II and Richard E. Stearns. 1968. Syntax-directed transduction. *Journal of the Association of Computing Machinery*, 15(3):465–488.

Yajuan Lü, Sheng Li, Tiejun Zhao, and Muyun Yang. 2002. Learning chinese bracketing knowledge based on a bilingual language model. In *COLING 2002 - Proceedings of the 19th International Conference on Computational Linguistics*.

I. Dan Melamed. 2003. Multitext grammars and synchronous parsers. In *Proceedings of NAACL/HLT*.

I. Dan Melamed. 2004. Statistical machine translation by parsing. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics: ACL 2004*, pages 653–660.

Günter Neumann. 1998. Interleaving natural language parsing and generation through uniform processing. *Artifical Intelligence*, 99:121–163.

Stuart Shieber. 1988. A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING), Budapest*.

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

dicate that our correspondence-guided approach is a promising alternative for an application context in which a word alignment is available.