Correspondence-guided Synchronous Parsing of Parallel Corpora

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

Abstract

We present an efficient dynamic programming algorithm for synchronous parsing of sentence pairs from a parallel corpus with a given word alignment. Unless there is a large proportion of words without a correspondence in the other language, the worst-case complexity is significantly reduced over standard synchronous parsing. The theoretical complexity results are corroborated by a quantitative experimental evaluation.

Our longer-term goal is to induce monolingual grammars from a parallel corpus, exploiting implicit information about syntactic structure obtained from correspondence patterns.[1] Here we provide an important prerequisite for parallel corpus-based grammar induction: an efficient algorithm for synchronous parsing, given a particular word alignment (e.g., the most likely option from a statistical alignment).

Synchronous grammars. We assume a straightforward extension of context-free grammars (compare the *transduction grammars* of [Lewis II and Stearns, 1968]): (1) the terminal and non-terminal categories are pairs of symbols (or NIL); (2) the sequence of daughters can differ for the two languages; we use a compact rule notation with a numerical ranking for the linear precedence in each language. The general form of a rule 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 (non-)terminal symbol for language L_1 and L_2, respectively, and i_l, j_l are natural numbers for the rank in the sequence for L_1 and L_2 (for NIL categories a special rank 0 is assumed). Compare fig. 1 for a sample analysis of the German/English sentence pair *Wir müssen deshalb die Agrarpolitik prüfen*/*So we must look at the agricultural policy*. We assume a normal form in which the right-hand side is ordered by the rank in L_1.[2] The formalism goes along with the **continuity assumption** that *every complete constituent is continuous in both languages*.[3]

Synchronous parsing. Our dynamic programming algorithm can be viewed as a variant of Earley parsing and generation, which again can be described by inference rules. For instance, the central *completion* step in Earley parsing can be described by the rule[4]

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

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). The natural way of generalizing context-free parsing to synchronous grammars is thus to use string indices in both dimensions. So we get inference rules like the following (there is another one in which the i_2/j_2 and j_2/k_2 indices are swapped between the two items above the line):

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

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 L_1 and L_2) [Wu, 1997; Melamed, 2003].

Correspondence-guided parsing. As an alternative to standard "rectangular indexing" we propose an asymmetric approach: one of the languages (L_1) provides the "primary index" – the string span in L_1 like in monolingual parsing. As a secondary index, L_2 contributes a chart-generation-style bit vector of the words covered, which is mainly used to guide parsing – i.e., certain options are eliminated. A complete sample index for *müssen*/*must* in fig. 1 would be $\langle [1,2], [00100000]\rangle$. Completion can be formulated as inference rule (3).[5] Condition (iii) excludes discontinuity in passive chart items, i.e., complete constituents; active items (i.e., partial constituents) may well contain discontinuities.

$$(3)\quad \frac{\langle X_1/X_2 \rightarrow \alpha \bullet\ Y_1{:}r_1/Y_2{:}r_2\ \ \beta, \langle [i,j], \mathbf{v}\rangle\rangle,\ \langle Y_1/Y_2 \rightarrow \gamma \bullet, \langle [j,k], \mathbf{w}\rangle\rangle}{\langle X_1/X_2 \rightarrow \alpha\ Y_1{:}r_1/Y_2{:}r_2 \bullet\ \beta, \langle [i,k], \mathbf{u}\rangle\rangle}$$

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

[1] Cp. the new PTOLEMAIOS project at Saarland University (http://www.coli.uni-saarland.de/~jonask/PTOLEMAIOS/).

[2] However, categories that are NIL in L_1 come last. If there are several, they are viewed as unordered with respect to each other.

[3] As [Melamed, 2003] discusses, such an assumption is empirically problematic with binary grammars. However, if flat analyses are assumed for clauses and NPs, the typical problematic cases are resolved.

[4] A chart item is specified through a position (•) in a production and a string span ($[l_1, l_2]$). $\langle X \rightarrow \alpha \bullet Y\beta, [i,j]\rangle$ is an active item recording that between position i and j, an incomplete X phrase has been found, which covers α, but still misses $Y\beta$. Items with a final • are called passive.

[5] We use the bold-faced variables $\mathbf{v}, \mathbf{w}, \mathbf{u}$ for bit vectors; OR performs bitwise disjunction on the vectors.

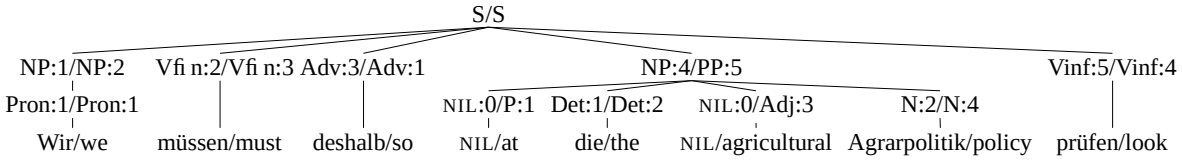


Figure 1: Sample analysis for a synchronous grammar

Parsing is successful if an item with index $\langle [0, N], \mathbf{1} \rangle$ can be found for the start category pair (where N is the length of the L_1-string).

Words in L_2 with no correspondent in L_1 (let's call them "L_1-NIL"s for short) 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 $\langle [i, i], [00100] \rangle$. But note that due to the continuity assumption, the distribution of the L_1-NILs is constrained by the other words in L_2, as exploited in the inference rule:

(4) $$\frac{\langle X_1/X_2 \rightarrow \alpha \bullet \text{NIL}{:}0/Y_2{:}r_2 \ \beta, \langle [i,j], \mathbf{v} \rangle, \quad \langle \text{NIL}/Y_2 \rightarrow \gamma \bullet, \langle [j,j], \mathbf{w} \rangle \rangle}{\langle X_1/X_2 \rightarrow \alpha \ \text{NIL}{:}0/Y_2{:}r_2 \ \bullet \ \beta, \langle [i,j], \mathbf{u} \rangle \rangle}$$ 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 after all the parts that have correspondents in both languages have been found.

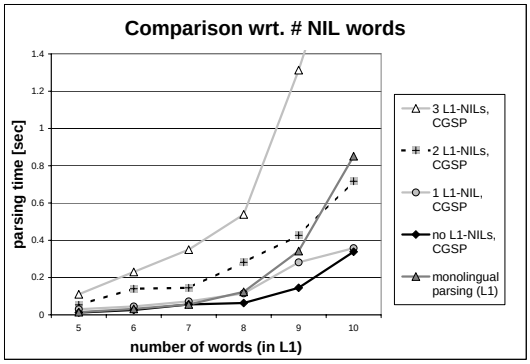
Complexity. We assume that the two-dimensional chart is initialized with the correspondences following from a word alignment. Hence, for each terminal that is non-empty in L_1, both components of the index are known. When two items with known secondary indices are combined with (3), the new secondary index can be determined with minimal expense. Thus, *for sentence pairs without any L_1-NILs*, the worst-case complexity for synchronous parsing is identical to the monolingual case of context-free parsing (i.e., $O(n^3)$). The average parsing expense in the absence of L_1-NILs is even lower than in monolingual parsing: certain hypotheses for complete constituents are excluded because the secondary index reveals a discontinuity.

The complexity is 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 (4)), the number of possible variants in the secondary index is a function of the number of L_1-NILs. Say there are m L_1-NILs. In each application of rule (4) we pick a vector $\mathbf{v}$, with a variable for the leftmost and rightmost L_1-NIL element. By adjacency, either the leftmost or rightmost one 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^3 expense of parsing non-nil words, we get an expense of m^3 for parsing the L_1-NILs, and end up in $O(n^3 m^3)$. Since typically the number of correspondent-less words is significantly lower than the total number of words, these results are encouraging for medium-to-large-scale grammar learning experiments.[6]

[6]The idealizing assumption of a single, deterministic word alignment is not realistic (as pointed out by a reviewer). Consideration of various possible alignments may lead to a worst-case combinatorial explosion. It is an empirical question for the future whether an effective heuristic can be found for narrowing down the space of alignments that have to be considered.

Empirical Evaluation. To validate empirically that the *average* parsing complexity for the proposed correspondence-guided synchronous parsing approach (CGSP) for sentences without or with few L_1-NILs is lower than for standard monolingual parsing, we did a prototype implementation of the algorithm and ran a comparison. A synchronous grammar was extracted (and smoothed) from a manually aligned German/English section of the Europarl corpus. The results are shown as the black line (for the CGSP approach on sentences without L_1-NILs) and dark gray line (for monolingual parsing) in (5). The diagram shows the average parsing time for sentence pairs of various lengths. Note that CGSP takes clearly less time.

(5) *Synchronous parsing with a growing number of L_1-NILs*



(5) also shows comparative results for parsing performance on sentences that do contain L_1-NILs (curves for 1, 2 and 3 L_1-NILs are shown). Here too, the theoretical results are corroborated that with a limited number of L_1-NILs, the CGSP is still efficient.

We also simulated a synchronous parser which does not take advantage of a given word alignment. 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).

References

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

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

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