Tree Linearization in English: Improving Language Model Based Approaches

Katja Filippova and **Michael Strube**
EML Research gGmbH
Schloss-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
`http://www.eml-research.de/nlp`

Abstract

We compare two approaches to dependency tree linearization, a task which arises in many NLP applications. The first one is the widely used 'overgenerate and rank' approach which relies exclusively on a trigram language model (LM); the second one combines language modeling with a maximum entropy classifier trained on a range of linguistic features. The results provide strong support for the combined method and show that trigram LMs are appropriate for phrase linearization while on the clause level a richer representation is necessary to achieve comparable performance.

1 Introduction

To date, many natural language processing applications rely on syntactic representations and also modify them by compressing, fusing, or translating into a different language. A syntactic tree emerging as a result of such operations has to be linearized to a string of words before it can be output to the end-user. The simple and most widely used trigram LM has become a standard tool for tree linearization in English (Langkilde & Knight, 1998). For languages with less rigid word order, LM-based approaches have been shown to perform poorly (e.g., Marsi & Krahmer (2005) for Dutch), and methods relying on a range of linguistic features have been successfully applied instead (see Uchimoto et al. (2000) and Ringger et al. (2004), Filippova & Strube (2007) for Japanese and German resp.). To our knowledge, none of the linearization studies have compared a LM-based method with an alternative. Thus, it would be of interest to draw such a comparison, especially on English data, where LMs are usually expected to work well.

As an improvement to the LM-based approach, we propose a combined method which distinguishes between the phrase and the clause levels:

- it relies on a trigram LM to order words within phrases;
- it finds the order of clause constituents (i.e., constituents dependent on a finite verb) with a maximum entropy classifier trained on a range of linguistic features.

We show that such a differentiated approach is beneficial and that the proposed combination outperforms the method which relies solely on a LM. Hence, our results challenge the widespread attitude that trigram LMs provide an appropriate way to linearize syntactic trees in English but also indicate that they perform well in linearizing subtrees corresponding to phrases.

2 LM-based Approach

Trigram models are easy to build and use, and it has been shown that more sophisticated n-gram models (e.g., with higher n, complex smoothing techniques, skipping, clustering or caching) are often not worth the effort of implementing them due to data sparseness and other issues (Goodman, 2001). This explains the popularity of trigram LMs in a variety of NLP tasks (Jurafsky & Martin, 2008), in particular, in tree linearization where they have become

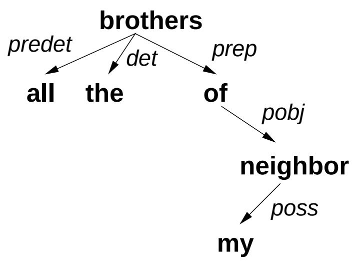

Figure 1: A tree of the noun phrase *all the brothers of my neighbor*

de facto the standard tree linearization tool in accordance with the 'overgenerate and rank' principle: given a syntactic tree, one needs to consider all possible linearizations and then choose the one with the lowest entropy. Given a projective dependency tree[1], all linearizations can be found recursively by generating permutations of a node and its children. Unfortunately, the number of possible permutations grows factorially with the branching factor. Hence it is highly desirable to prohibit generation of clearly unacceptable permutations by putting hard constraints encoded in the English grammar. The constraints which we implement in our study are the following: determiners, possessives, quantifiers and noun or adjective modifiers always **precede** their heads. Conjunctions, coordinated elements, prepositional objects always **follow** their heads. These constraints allow us to limit, e.g., the total of 96 ($2 \times 2 \times 4!$) possibilities for the tree corresponding to the phrase *all the brothers of my neighbor* (see Figure 1) to only two (*all the brothers of my neighbor, the all brothers of my neighbor*).

Still, even with such constraints, in some cases the list of possible linearizations is too long and has to be reduced to the first N, where N is supposed to be sufficiently large. In our experiments we break the permutation generation process if the limit of 20,000 variants is reached.

3 Combined Approach

The LM approach described above has at least two disadvantages: (1) long distance dependencies are not captured, and (2) the list of all possible linearizations can be huge which makes the search for the best string unfeasible. However, our combined approach is based on the premise that trigram LMs are well-suited for finding the order within NPs, PPs and other phrases where the head is not a finite verb. E.g., given a noun modified by the words *big*, *red* and *the*, a LM can reliably rank the correct order higher than incorrect ones (*the big red N* vs. *the red big N*, etc.).

Next, on the clause level, for every finite verb in the tree we find the order of its dependents using the method which we originally developed for German (Filippova & Strube, 2007), which utilizes a range of such linguistic features as PoS tag, syntactic role, length in words, pronominalization, semantic class, etc.[2] For the experiments presented in this paper, we train two maximum entropy classifiers on all but the semantic features:

1. The first classifier determines the best starting point for a sentence: for each constituent dependent on the verb it returns the probability of this constituent being the first one in a sentence. The subject and also adjuncts (e.g. temporal adjuncts like *yesterday*) are usually found in the beginning of the sentence.

2. The second classifier is trained to determine whether the precedence relation holds between two adjacent constituents and is applied to all constituents but the one selected by the first classifier. The precedence relation defined by this classifier has been shown to be transitive and thus can be used to sort randomly ordered constituents. Note that we do not need to consider all possible orders to find the best one.

Once the order within clause constituents as well as the order among them is found, the verb is placed right after the subject. The verb placing step completes the linearization process.

The need for two distinct classifiers can be illustrated with the following example:

(1) a [*Earlier today*] [she] sent [him] [an email].

b [She] sent [him] [an email] [*earlier today*].

c *[She] sent [*earlier today*] [him] [an email].

[1]Note that a phrase structure tree can be converted into a dependency tree, and some PCFG parsers provide this option.

[2]See the cited paper for the full list of features and implementation details.

(1a,b) are grammatical while (1c) is hardly acceptable, and no simple precedence rule can be learned from pairs of constituents in (1a) and (1b): the temporal adjunct *earlier today* can precede or follow each of the other constituents dependent on the verb (*she, him, an email*). Thus, the classifier which determines the precedence relation is not enough. However, an adequate rule can be inferred with an additional classifier trained to find good starting points: a temporal adjunct may appear as the first constituent in a sentence; if it is not chosen for this position, it should be preceded by the pronominalized subject (*she*), the indirect object (*him*) and the short non-pronominalized object (*an email*).

4 Experiments

The goal of our experiments is to check the following hypotheses:

1. That trigram LMs are well-suited for phrase linearization.
2. That there is a considerable drop in performance when one uses them for linearization on the clause level.
3. That an approach which uses a richer representation on the clause level is more appropriate.

4.1 Data

We take a subset of the TIPSTER[3] corpus – all Wall Street Journal articles from the period of 1987-92 (approx. 72 mill. words) – and automatically annotate them with sentence boundaries, part of speech tags and dependency relations using the Stanford parser (Klein & Manning, 2003). We reserve a small subset of about 600 articles (340,000 words) for testing and use the rest to build a trigram LM with the CMU toolkit (Clarkson & Rosenfeld, 1997, with Good-Turing smoothing and vocabulary size of 30,000). To train the maximum entropy classifiers we use about 41,000 sentences.

4.2 Evaluation

To test the trigram-based approach, we generate all possible permutations of clause constituents, place the verb right after the subject and then rank the resulting strings with the LM taking the information on sentence boundaries into account. To test the combined approach, we find the best candidate for the first position in the clause, then put the remaining constituents in a random order, and finally sort them by consulting the second classifier.

The purpose of the evaluation is to assess how good a method is at reproducing the input from its dependency tree. We separately evaluate the performance on the phrase and the clause levels. When comparing the two methods on the clause level, we take the clause constituents as they are presented in the input sentence. Although English allows for some minor variation in word order and it might happen that the generated order is not necessarily wrong if different from the original one, we do not expect this to happen often and evaluate the performance rigorously: only the original order counts as the correct one. The default evaluation metric is per-phrase/per-clause accuracy:

$$acc = \frac{|correct|}{|total|}$$

Other metrics we use to measure how different a generated order of N elements is from the correct one are:

1. Kendall's τ, $\tau = 1 - 4\frac{t}{N(N-1)}$ where t is the minimum number of interchanges of consecutive elements to achieve the right order (Kendall, 1938; Lapata, 2006).
2. Edit distance related di, $di = 1 - \frac{m}{N}$ where m is the minimum number of deletions combined with insertions to get to the right order (Ringger et al., 2004).

E.g., on the phrase level, the incorrectly generated phrase *the all brothers of my neighbor* ('1-0-2-3-4-5') gets $\tau = 0.87$, $di = 0.83$. Likewise, given the input sentence from (1a), the incorrectly generated order of the four clause constituents in (1c) – '1-0-2-3' – gets τ of 0.67 and di of 0.75.

4.3 Results

The results of the experiments on the phrase and the clause levels are presented in Tables 1 and 2 respectively. From the total of 5,000 phrases, 55 (about

[3]Description at `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93T3A`.

1%) were discarded because the number of admissible linearizations exceeded the limit of 20,000. In the first row of Table 1 we give the results for cases where, with all constraints applied, there were still several possible linearizations (*non-triv*; 1,797); the second row is for all phrases which were longer than one word (> 1; 2,791); the bottom row presents the results for the total of 4,945 phrases (*all*).

	acc	τ	*di*
non-triv	76%	0.85	0.94
> 1	85%	0.90	0.96
all	91%	0.94	0.98

Table 1: Results of the trigram method on the phrase level

Table 2 presents the results of the trigram-based (TRIGRAM) and combined (COMBINED) methods on the clause level. Here, we filtered out trivial cases and considered only clauses which had at least two constituents dependent on the verb (approx. 5,000 clauses in total).

	acc	τ	*di*
TRIGRAM	49%	0.49	0.81
COMBINED	67%	0.71	0.88

Table 2: Results of the two methods on the clause level

4.4 Discussion

The difference in accuracy between the performance of the trigram model on the phrase and the clause level is considerable – 76% vs. 49%. The accuracy of 76% is remarkable given that the average length of phrases which counted as *non-triv* is 6.2 words, whereas the average clause length in constituents is 3.3. This statistically significant difference in performance supports our hypothesis that the 'overgenerate and rank' approach advocated in earlier studies is more adequate for finding the optimal order within phrases. The τ value of 0.85 also indicates that many of the wrong phrase linearizations were near misses. On the clause level, where long distance dependencies are frequent, an approach which takes a range of grammatical features into account is more appropriate – this is confirmed by the significantly better results of the combined method (67%).

5 Conclusions

We investigated two tree linearization methods in English: the mainstream trigram-based approach and the one which combines a trigram LM on the phrase level with two classifiers trained on a range of linguistic features on the clause level. The results demonstrate (1) that the combined approach reproduces the word order more accurately, and (2) that the performance of the trigram LM-based method on phrases is significantly better than on clauses.

Acknowledgments: This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first author has been supported by a KTF grant (09.009.2004). We would like to thank the anonymous reviewers for their feedback.

References

Clarkson, P. & R. Rosenfeld (1997). Statistical language modeling using the CMU-Cambridge toolkit. In *Proc. of EUROSPEECH-97*, pp. 2707–2710.

Filippova, K. & M. Strube (2007). Generating constituent order in German clauses. In *Proc. of ACL-07*, pp. 320–327.

Goodman, J. T. (2001). A bit of progress in language modeling. *Computer Speech and Language*, pp. 403–434.

Jurafsky, D. & J. H. Martin (2008). *Speech and Language Processing*. Upper Saddle River, N.J.: Prentice Hall.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30:81–93.

Klein, D. & C. D. Manning (2003). Accurate unlexicalized parsing. In *Proc. of ACL-03*, pp. 423–430.

Langkilde, I. & K. Knight (1998). Generation that exploits corpus-based statistical knowledge. In *Proc. of COLING-ACL-98*, pp. 704–710.

Lapata, M. (2006). Automatic evaluation of information ordering: Kendall's tau. *Computational Linguistics*, 32(4):471–484.

Marsi, E. & E. Krahmer (2005). Explorations in sentence fusion. In *Proc. of ENLG-05*, pp. 109–117.

Ringger, E., M. Gamon, R. C. Moore, D. Rojas, M. Smets & S. Corston-Oliver (2004). Linguistically informed statistical models of constituent structure for ordering in sentence realization. In *Proc. of COLING-04*, pp. 673–679.

Uchimoto, K., M. Murata, Q. Ma, S. Sekine & H. Isahara (2000). Word order acquisition from corpora. In *Proc. of COLING-00*, pp. 871–877.