Automatic Alignment of Czech and English Deep Syntactic Dependency Trees*

David Mareček, Zdeněk Žabokrtský, and Václav Novák

Institute of Formal and Applied Linguistics,
Charles University in Prague, Czech Republic
david.marecek@gmail.com, {zabokrtsky, novak}@ufal.mff.cuni.cz

Abstract. In this paper, we focus on alignment of Czech and English tectogrammatical dependency trees. The alignment of deep syntactic dependency trees can be used for training transfer models for machine translation systems based on analysis-transfer-synthesis architecture. The results of our experiments show that shifting the alignment task from the word layer to the tectogrammatical layer both (a) increases the inter-annotator agreement on the task and (b) allows to construct a feature-based algorithm which uses sentence structure and which outperforms the GIZA++ aligner in terms of f-measure on aligned tectogrammatical node pairs.

1 Introduction

Statistical machine translation requires a substantial amount of translation knowledge typically acquired from parallel corpora. For training a transfer on deeper syntactic layer it is feasible to use aligned deep syntactic trees.

At first we will show the differences between alignment of sentences on the surface (word alignment) and alignment of their tectogrammatical representations – deep syntactic dependency trees according to the specification of Prague Dependency Treebank 2.0 [1]. Tectogrammatical trees (t-trees for short, analogously t-layer and t-nodes) will be described in Section 2.

The word alignment task is to find the most likely counterpart for every word in a sentence. It is questionable if we really need to find counterparts for all words, especially in the case of typologically different languages. For example, auxiliary words in one language differ in their functions and repertory from auxiliary words in another one.

There is an example of English sentence and its Czech translation in Figure 1. The full arrows represent the obvious alignment pairs, whereas the correspondence expressed by the dashed arrows is not straightforward. For example, there is only one negation word *No* in the English sentence while in the Czech one, there is the negation in both *Žádné* and *nebylo*. The word *nebylo* can be translated into English as *wasn't*, but if the word *dosud* follows, the only possibility

* The research reported on in this paper has been supported by the projects GAČR 405/06/0589, 1ET101120503, MSM0021620838, 1ET201120505, and LC 536.

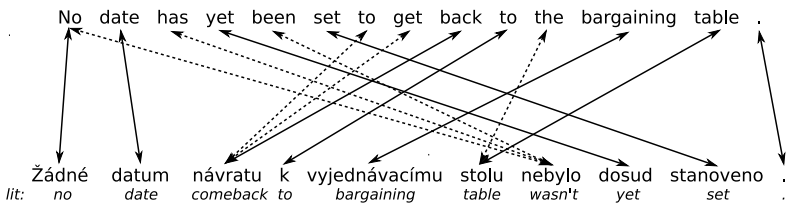

Fig. 1. Example of word alignment on the surface.

is present perfect tense – *has been*. The word *dosud* has thus a relationship with the present perfect tense and should be linked besides *yet* also with *has* and *been*. This example illustrates the fact that the word-alignment of Czech-English sentence pairs is rather complex. [2] describe an experiment in which two annotators aligned manually 515 sentences from Czech-English corpus. The inter-annotator agreement of the simplest word alignment method (only one type of edge) reached 91%.

In this paper we will be concerned with alignment of deep syntactic t-trees. On the t-layer the Czech and English sentence trees are more similar compared to the similarity of their surface shapes. Alignment on t-layer for the same sentence as in Figure 1 is shown in Figure 2 (the t-tree visualization is highly simplified: only t-lemmas are depicted with the t-nodes). We can see that the alignment pairs made in t-trees are exactly those that were aligned as evident (full arrows) on the surface.

In the reported experiments, we are interested in the t-layer alignment, not in the word alignment. We transform the manual word-layer alignment up to the t-layer (which results more or less in choosing only the links between content words). We show that the inter-annotator agreement on such t-layer links is

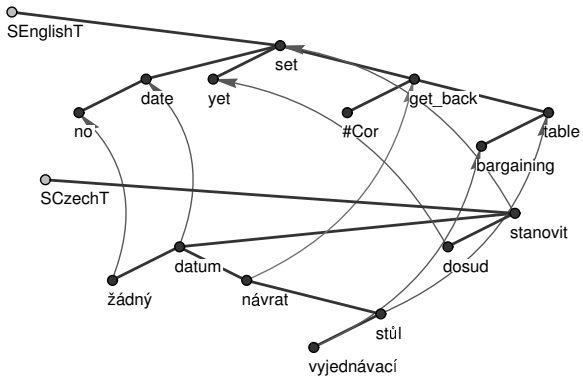

Fig. 2. Example of alignment on the t-layer (t-trees are simplified).

higher compared to that on word-layer. The main result achieved in the presented work is following: we show that the t-alignment produced by our feature-based structural t-aligner outperforms the t-alignment derived from the word-layer alignment provided by GIZA++ [3] in terms of their f-measure. This is probably caused by the fact that tectogrammatical representations of Czech and English sentences are much closer compared to the distance of their surface shapes. For example, most auxiliary words, whose alignment is notoriously problematic, are not represented as tectogrammatical nodes on their own and thus no artificial rules for their alignment are needed.

Nodes of t-trees represent content words in sentences. Haruno and Yamazaki were engaged in alignment of content words only for Japanese-English pair [4], with the motivation similar to ours: it is not feasible to align functional words in structurally very different languages; however, they did not use tree structures. Experiments with alignment of dependency trees are described for example in [5] and in [6], but in our opinion no quantitative comparison of these approaches with our approach is possible due to different experiment contexts. There is also a broad literature about aligning constituency trees; dependency and constituency approaches to alignment are compared in [7].

2 Tectogrammatical representation

The tectogrammatical representation is based on the Functional Generative Description, developed by Petr Sgall and his collaborators since 1960s (see [8]). We use its implementation specified in Prague Dependency Treebank 2.0 as described in [1]. It consists of three interlinked annotation layers: the morphological layer, the analytical layer (a-layer for short, describing the surface syntax) and the tectogrammatical layer (t-layer, describing the deep syntax – transition between syntax and semantics).

On the t-layer, every sentence is represented as a rooted dependency tree. Unlike in the case of the analytical layer, only auto-semantic (content) words have their own nodes in the t-trees. Function words like auxiliary verbs, subordinating conjunctions and prepositions are represented in the respective nodes in the form of their attributes. For example, there is no node representing auxiliary *will* at the t-layer, but its meaning is captured by attribute *tense*. Other attributes describe several cognitive, syntactic and morphological categories. The presence of an attribute in a node is determined by the node type. In this paper we will use the following attributes:

- *t-lemma* – tectogrammatical lemma,
- *formeme* – simplified description of the morphosyntactic form (how the t-node is expressed on the surface), introduced in [9]
- *deepord* – describes the organization of words in a sentence according to their increasing communicative dynamism. It also determines the linear position of t-node in the tree.

As for the alignment on t-layer we currently do not distinguish different arrow types. Every t-node is aligned with no, one or more t-nodes in the opposite

language. Sometimes it is necessary to align more Czech t-nodes with more English t-nodes. Then the arrows will lead from each such Czech t-node to all corresponding English t-nodes. This occurs infrequently, typically only at idioms.

3 Preprocessing and GIZA++ Alignment

We used parallel texts from Prague Czech-English Dependency Treebank, PCEDT for short [10]. PCEDT contains Czech translations of 21,600 English sentences from the Wall Street Journal part of Penn Treebank corpus. 515 sentence pairs were also aligned on word layer [2] and we will use them for evaluating our t-aligner.

All sentences were automatically parsed up to the t-layer using TectoMT, software framework for developing machine translation systems [9]. Czech sentences were morphologically analyzed and disambiguated by the morphological tagger shipped with PDT2.0 [1], then syntactically analyzed by McDonald's MST parser [11], and automatically converted into t-trees. English sentences were tagged by the TnT tagger [12] and analyzed by the Collins parser [13]; phrase-structure trees were then converted into a-trees and finally into the t-trees.

We aligned t-trees by GIZA++ tool [3]. The sequences of *t-lemmas* were extracted form all the trees, ordered according to their *deepord* attribute and given as the input to the GIZA++ tool. Note that there is no information about the tree structure or other attributes. However, the accuracy of this alignment reached 83% (we will describe the evaluation method in Section 7). Thus we decided to include alignment generated by GIZA++ into the input for our t-aligner. We use also the probability table generated by GIZA++.

4 Greedy Algorithm for 1:1 Alignment

The alignment process consists of two phases. In the first phase feature-based greedy algorithm aligns trees. There are only 1:1 alignments allowed (each t-node can have at most one counterpart). In the second phase simple algorithm finds unaligned t-nodes and tries to align them with already aligned t-nodes in the opposite language.

The first phase is based on a linear model and was inspired by [5]. First, all potential alignment pairs between two trees are considered. To each such pair (e_i, c_j) we assign its score which is computed as: $S(c_i, e_j) = \overrightarrow{w} \cdot \overrightarrow{f}(c_i, e_j)$, where c_i is the $i-$th Czech t-node, e_j is the $j-$th English t-node, $\overrightarrow{w}$ is the vector of feature weights, and $\overrightarrow{f}$ is the vector of feature values. The features are listed in Section 5. All features were designed manually.

Pseudo-code of this algorithm is given in Figure 3. In each iteration a pair with the best score is aligned, which is repeated as long as both t-trees contain unaligned t-nodes and the best pair score is higher than a threshold. It is necessary to update pair scores after each step, because some features might be influenced by the already aligned pairs.

```
Input: TreePairs – Czech and English t-tree pairs
Output: Aligned trees
foreach (CT, ET) ∈ TreePairs do
    foreach cnode ∈ CT do
        used(cnode) = 0;
        foreach enode ∈ ET do
            used(enode) = 0;
            score(cnode, enode) = w⃗ · f⃗(cnode, enode);
    while ∃(cnode, enode): used(cnode) = 0 and used(enode) = 0 do
        Find (cmax, emax) with the highest score(cmax, emax);
        if score(cmax, emax) ≥ threshold then
            Align(cmax, emax);
            used(cmax) = 1;
            used(emax) = 1;
            foreach cnode ∈ CT, enode ∈ ET do
                if used(cnode) = 0 and used(enode) = 0 then
                    if cnode = parent(cmax) or cnode ∈ children(cmax)
                    or enode = parent(emax) or enode ∈ children(emax) then
                        score(cnode, enode) = w⃗ · f⃗(cnode, enode);
        else
            Break;
```

Fig. 3. Pseudo-code for the first phase of t-alignment

We used an implementation of the discriminative reranker described in [14] (basically a modification of averaged perceptron) for optimizing the feature weights.

5 Features

Features are individual measurable properties of a pair of Czech and English t-nodes. They include similarities of t-lemmas and other attributes of t-nodes, position in trees and they also take into account whether GIZA++ did align this pair or not.

Several features use translation probabilities acquired either from probabilistic dictionary which is included in PCEDT corpus [10], or from tables of translation probabilities generated by GIZA++.

Features can return binary, integer, or real values. We used the following features:

- **t-lemma pair in dictionary** (*binary*) – Equal to 1 if the pair of t-lemmas occurs in the translation dictionary (otherwise equal to 0).
- **translation probability from dictionary** (*real*) – Returns an unidirectional t-lemma translation probability from English to Czech contained in the dictionary: $p_{dict}(e_i, c_j) = p(t\_lemma(e_i) \mid t\_lemma(c_j))$
- **aligned by GIZA++, intersection** (*binary*) – Equal to 1 if the two nodes were aligned by GIZA++ with the intersection symmetrization.

- **aligned by GIZA++, grow-diag-final** (*binary*) – Equal to 1 if the two nodes were aligned by GIZA++ with the grow-diag-final symmetrization.
- **translation probability from GIZA++** (*real*) – Returns the mean of t-lemma translation probabilities in both directions that were acquired from GIZA++ output translation tables.
- **identical t-lemmas** (*binary*) – Equal to 1 if Czech t-lemma is the same string as the English one.
- **5 letter match** (*binary*) – Equal to 1 if the five-letter prefixes of Czech and English t-lemmas are identical.
- **4 letter match** (*binary*) – Equal to 1 if the four-letter prefixes of Czech and English t-lemmas are identical and five-letter prefixes are not.
- **3 letter match** (*binary*) – Equal to 1 if the three-letter prefixes of Czech and English t-lemmas are identical and four-letter prefixes are not.
- **equal number prefix** (*binary*) – Equal to 1 if both Czech and English t-lemmas start with the same sequence of digits.
- **aligned parent** (*binary*) – Equal to 1 if the parent of Czech t-node is already aligned with the parent of English t-node.
- **aligned child** (*integer*) – Number of Czech t-node children that are already aligned with children of English t-node.
- **both coap** (*binary*) – Equal to 1 if both t-nodes are roots of coordination or apposition constructions.
- **same shortened formeme** (*binary*) – Every formeme contains information about the semantic part of speech it can be applied to (e.g., *n*, *v*, *adj* or *adv*). This feature equals to 1 if both semantic parts of speech are equal.
- **similarity in linear position** (*real*) – Linear position of each t-node is stored in its attribute *deepord*. As for similarity, we can compute the difference between relative positions of correspondent t-nodes and subtract it form 1. The numbers $|c|$ and $|e|$ denote counts of t-nodes in Czech and English t-trees. $sim(e_i, c_j) = 1 - \left|\frac{i}{|e|} - \frac{j}{|c|}\right|$

6 Completing 1:N Alignments

In this phase, the algorithm goes through all the t-nodes that have not been aligned yet. If a t-node K is not aligned and its parent t-node $P(K)$ is aligned to a node L in the opposite language, we denote the pair $K - L$ as a candidate pair. Similarly, if the unaligned t-node M has a child t-node $C(M)$ which is aligned to a t-node N, $M - N$ becomes a candidate pair too.

If the candidate pair was aligned also by GIZA++ with the grow-diag-final symmetrization method and this pair also exists in the PCEDT dictionary, we align this pair of t-nodes.

7 Evaluation

For evaluating the t-aligner, we used 515 sentences (about 13,000 tokens) from the Czech-English corpus PCEDT [10]. The sentences were manually aligned on

the word layer by two annotators [2]. The annotators were asked to used three types of connections:

- *sure link* is used when individual words match,
- *phrasal link* when whole phrases correspond but not words by themselves,
- *possible links* connect words that do not have a real equivalent in the other language, but syntactically clearly belong to a contact word nearby, such as English articles.

The inter-annotator agreement on the word level is 82%. If we disregard the types of connection, the agreement reaches 91%.

To measure the inter-annotator agreement on the t-layer, we need to transform the word alignment as follows: Every t-node has an attribute which can point to one word on the surface from which it got its lexical meaning. Thus two t-nodes are aligned, if their corresponding words on word level are aligned.

Inter-annotator agreement of the alignment transferred to t-layer reached 94.7% (we do not distinguish the types of connections). The increase in agreement shows that some of the problematic words on the surface are not represented as t-nodes of their own. This percentage can be considered as an upper limit for the t-aligner accuracy.

We combined the two parallel annotations to one gold-standard alignment according to the following rules: a connection is marked as sure if at least one of the annotators marked it as sure and the other also supported the link by any connection type. In all other cases (at least one annotator makes any type of link), the connection is marked as possible.

There are two possibilities how to deal with the golden alignment. There are two types of connections – *sure* and *possible* one, while our structural t-aligner makes only one type of connection. We work with two evaluation variants: (1) **both types** – we take both types of connections as equivalent and compare them with connections made by t-aligner, (2) **sure only** – we take only the *sure* connections and compare them with connections made by t-aligner

8 Experiments and Results

First, we evaluated our structural t-aligner. 10-fold cross-validation was used because of the data size (only 515 sentences).

Precision, recall and f-measure were computed in each iteration. Precision indicates the percentage of how many pairs aligned by this algorithm were aligned

Table 1. 10-fold cross-validation results of the structural t-aligner (*sure only*).

n	1	2	3	4	5	6	7	8	9	10	mean	σ
P	93.63	93.40	93.52	95.49	88.65	92.83	95.22	92.12	92.19	93.72	93.08	1.91
R	87.92	89.78	90.66	91.25	80.30	87.43	89.47	89.88	82.58	90.14	87.94	3.65
F	90.69	91.55	92.07	93.32	84.27	90.04	92.25	90.98	87.12	91.90	**90.42**	**2.73**

also by annotator; recall indicates how many pairs aligned by the annotator were aligned by the algorithm. F-measure is their harmonic mean ($f = 2pr/(p + r)$).

The results are averaged and the standard deviation is computed in Table 1, using the *sure only* evaluation variant. The mean f-measure of the structural t-aligner is 90.42%. The comparison with the *both types* variant is in Table 2.

Table 2. Evaluation of the structural t-aligner.

evaluation variant	precision	recall	f-measure
sure only	93.08	87.94	90.42
both types	95.20	80.16	87.04

We also measured precision, recall and f-measure of the GIZA++ tool on sequences of t-lemmas (described in Section 3). GIZA++ output is asymmetric. It makes at most one connection from one word from source language into target language. We ran GIZA++ in both directions and symmetrize the two outputs. We used three symmetrization methods: (1) **intersection** – only connections that occurs in both outputs are used, (2) **union** – connections that occurs in at least one output are used, (3) **grow-diag-final** – something between intersection and union, see [3] for details. Evaluation of the t-alignment by GIZA++ is presented in Table 3. Again, we used the *sure only* evaluation variant.

Table 3. *Sure only* evaluation of t-alignment by GIZA++

symmetrization	precision	recall	f-measure
intersection	93.3	74.2	82.6
union	64.8	89.6	75.2
grow-diag-final	71.9	87.3	78.9

The vector of feature weights (resulting from one iteration of the cross validation) is shown in Table 4. Besides the weight vector, also the threshold value is needed in the algorithm. We found it by hill-climbing method after the feature weights were estimated. Its optimal value for weights given in Table 4 is 3.42.

There is another example of aligned trees in Figure 8. In this case, our t-aligner made no errors (even if the t-trees contain some errors). We can see that most arrows are more or less vertical. This implies relatively high weight of the feature "similarity in linear position". The pair *brokerage – makléřský* is not in the dictionary, but the "aligned parent" feature can help to choose the appropriate alignment. The pair *margin – maržní* is also not present in the dictionary and parents are not aligned. In this case the feature "3 letter match" can be helpful.

Table 4. Feature weights obtained by the perceptron

feature	values	weight
similarity in linear position	⟨ 0, 1 ⟩	2.81
aligned by Giza, intersection	0 or 1	2.78
equal number prefix	0 or 1	2.63
5 letter match	0 or 1	2.28
4 letter match	0 or 1	1.81
translation probability from Giza	⟨ 0, 1 ⟩	1.49
identical t-lemmas	0 or 1	1.00
t-lemma pair in dictionary	0 or 1	0.95
aligned by Giza, grow-diag-final	0 or 1	0.64
both coap	0 or 1	0.51
3 letter match	0 or 1	0.49
aligned parent	0 or 1	0.37
aligned child	0, 1, 2, 3,...	0.33
translation probability from dict.	⟨ 0, 1 ⟩	0.17
same shortened formeme	0 or 1	0.11

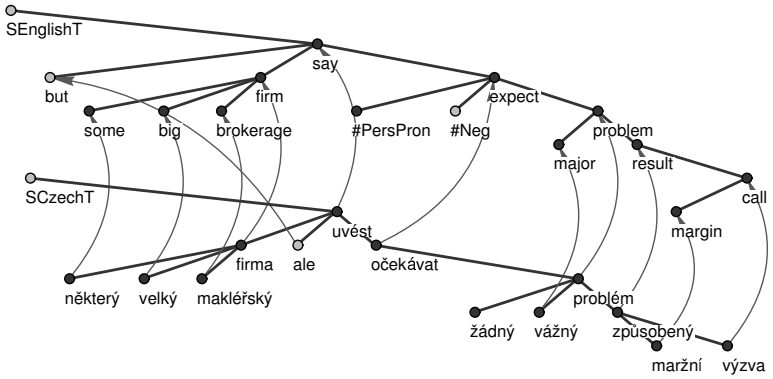

Fig. 4. T-tree alignment of the sentences *"But some big brokerage firms said they don't expect major problems as a result of margin calls."* and *"Některé velké makléřské firmy ale uvedly, že neočekávají žádné vážné problémy způsobené maržními výzvami."*

9 Conclusions

We have presented an algorithm for aligning t-trees based on manually designed features. The weights of the features were trained by a perceptron-based reranker. This algorithm also uses an alignment made by GIZA++. The feature weights show that the linear position of a t-node in the tree is the most important feature, but the structural and lexical features help too. The resulting f-measure of our t-aligner is 90.4%. This result is still well below the upper limit – the inter-annotator agreement on t-layer alignment reaches 94.7% –, but sig-

nificantly outperforms the 82.6% baseline t-alignment derived from the GIZA++ word alignment.

The most problematic relations are those which are not 1:1. The second phase of our t-aligner makes only a few 1:N connections. We do not deal with N:N connections at all, although they exist in our evaluation data.

References

1. Hajič, J., Hajičová, E., Panevová, J., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M.: Prague Dependency Treebank 2.0. Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia (2006)
2. Bojar, O., Prokopová, M.: Czech-English Word Alignment. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006), ELRA (May 2006) 1236–1239
3. Och, F.J., Ney, H.: A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics **29**(1) (2003) 19–51
4. Haruno, M., Yamazaki, T.: High-performance Bilingual Text Alignment Using Statistical and Dictionary Information. In: Proceedings of the 34th conference of the Association for Computational Linguistics. (1996) 131–138
5. Menezes, A., Richardson, S.D.: A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In: Proceedings of the workshop on Data-driven methods in machine translation. Volume 14. (2001) 1–8
6. Bojar, O., Cinková, S., Ptáček, J.: Towards English-to-Czech MT via Tectogrammatical Layer. In: Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories (TLT 2007), Bergen, Norway (December 2007)
7. Gildea, D.: Dependencies vs. constituents for tree-based alignment. In Lin, D., Wu, D., eds.: Proceedings of EMNLP 2004, Barcelona, Spain, Association for Computational Linguistics (July 2004) 214–221
8. Sgall, P.: Generativní popis jazyka a česká deklinace. Academia, Prague, Czech Republic (1967)
9. Žabokrtský, Z., Ptáček, J., Pajas, P.: TectoMT: Highly Modular MT System with Tectogrammatics Used as Transfer Layer. In: Proceedings of the 3rd Workshop on Statistical Machine Translation, ACL. (2008)
10. Cuřín, J., Čmejrek, M., Havelka, J., Hajič, J., Kuboň, V., Žabokrtský, Z.: Prague Czech-English Dependency Treebank, Version 1.0. Linguistics Data Consortium, Catalog No.: LDC2004T25 (2004)
11. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-Projective Dependency Parsing using Spanning Tree Algorithms. In: Proceedings of Human Langauge Technology Conference and Conference on Empirical Methods in Natural Language Processing (HTL/EMNLP), Vancouver, BC, Canada (2005) 523–530
12. Brants, T.: TnT - A Statistical Part-of-Speech Tagger. In: Proceedings of the 6th Applied Natural Language Processing Conference, Seattle (2000) 224–231
13. Collins, M.: Head-driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania, Philadelphia (1999)
14. Collins, M.: Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In: Proceedings of EMNLP. Volume 10. (2002) 1–8