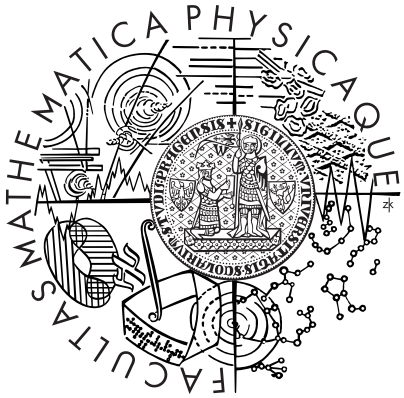
Charles University in Prague
Faculty of Mathematics and Physics

DIPLOMA THESIS



David Mareček

Automatic Alignment of Tectogrammatical Trees from Czech-English Parallel Corpus

Institute of Formal and Applied Linguistics

Supervisor:	Ing. Zdeněk Žabokrtský, Ph.D.
Study programme:	Computer Science
Study field:	Mathematical Linguistics

Prague, 2008

Acknowledgements

I thank my supervisor, Ing. Zdeněk Žabokrtský, Ph.D., for many insightful conversations during the development of the ideas in this thesis, and for helpful comments on the text.

Many thanks go to annotators – Martin Popel, Zuzana Škardová, Jiří Januška, and Dr. phil. Markus Giger – for careful work on word alignment and for their numerous helpful suggestions.

I would like to thank my family and my girlfriend Pavlína for their support throughout the duration of work on this thesis.

I certify that this diploma thesis is all my own work, and that I used only the cited literature. The thesis is freely available for all who can use it.

Prague, July 8, 2008 David Mareček

Title: Automatic Alignment of Tectogrammatical Trees from Czech-English Parallel Corpus
Author: David Mareček
Department: Institute of Formal and Applied Linguistics
Supervisor: Ing. Zdeněk Žabokrtský, Ph.D.
Supervisor's e-mail address: zabokrtsky@ufal.mff.cuni.cz

Abstract: The goal of this thesis is to implement and evaluate a software tool for automatic alignment of Czech and English tectogrammatical trees. The task is to find correspondent nodes between two trees that represent an English sentence and its Czech translation. Great amount of aligned trees acquired from parallel corpora can be used for training transfer models for machine translation systems. It is also useful for linguists in studying translation equivalents in two languages. In this thesis there is also described word alignment annotation process. The manual word alignment was necessary for evaluation of the aligner. 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. 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.

Keywords: tectogrammatical trees, word alignment, machine translation

Název práce: Automatické párování tektogramatických stromů z česko-anglického paralelního korpusu
Autor: David Mareček
Katedra (ústav): Ústav formální a aplikované lingvistiky
Vedoucí diplomové práce: Ing. Zdeněk Žabokrtský, Ph.D.
E-mail vedoucího: zabokrtsky@ufal.mff.cuni.cz

Abstrakt: Cílem této práce je implementovat a zhodnotit softwarový nástroj pro automatické zarovnávání (alignment) českých a anglických tektogramatických stromů. Úkolem je najít odpovídající si uzly stromů, které reprezentují anglickou větu a její český překlad. Velké množství zarovnaných stromů získaných z paralelního korpusu může být užitečné pro trénování modelu pro transfer strojového překladu. Zároveň může posloužit lingvistům při studování překladových ekvivalentů mezi dvěma jazyky. Výsledky našich experimentů ukazují, že přesunutím problému alignmentu ze slovní roviny na tektogramatickou (a) zvýšíme mezianotátorskou shodu (b) můžeme vytvořit alignovací algoritmus, který využívá i stromovou strukturu věty a překoná nástroj pro alignment GIZA++ spuštěný na uzly tektogramatických stromů. To je pravděpodobně zapříčiněno tím, že tektogramatické reprezentace českých a anglických vět si jsou mnohem podobnější než samotné věty na povrchu.

Klíčová slova: tektogramatická rovina, word alignment, strojový překlad

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Statistical machine translation requires a substantial amount of translation knowledge typically acquired from parallel corpora. We will focus on the machine translation system based on the analysis-transfer-synthesis architecture with the transfer on the deep syntactic layer. For the transfer step it is feasible to use either a great amount of aligned tree pairs or a large lexicon comprising not only dictionary word pairs and their translation probabilities, but also adequate amount of longer phrases translations. It is like the "chicken and egg" problem. If we have a good alignment, we can simply generate a large probabilistic lexicon. Reversely, with such a lexicon it is no problem to make the alignment.

In this thesis we will be concerned with alignment of Czech-English tectogrammatical trees – deep syntactic dependency trees according to the specification of Prague Dependency Treebank 2.0 [Hajič et al., 2006]. Tectogrammatical trees (t-trees for short) will be described in Section 3.1.

1.1 Why Tectogrammatical Trees?

At first we will show the differences between alignment of sentences on the surface (word alignment) and alignment of their tectogrammatical representations. 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 1.1. 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 is present perfect tense – *has been*. The word *dosud* has thus a relationship with the present

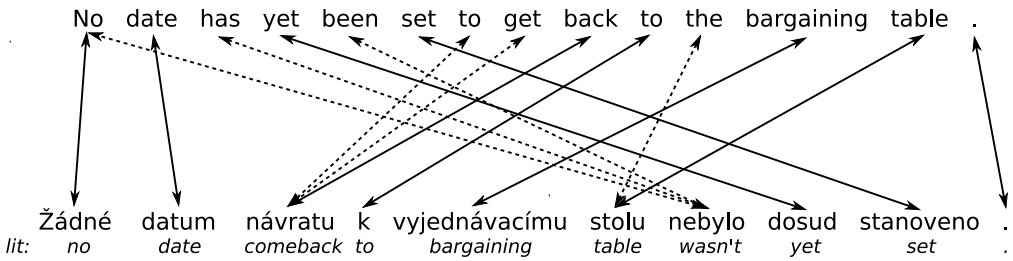


Figure 1.1: Example of word alignment on the surface.

perfect tense and should be linked besides *yet* also with *has* and *been*. This illustrates that word-alignment for Czech-English sentence pairs is rather complex. [Bojar and Prokopová, 2006] 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 the tectogrammatical layer the Czech and English sentence trees are more similar compared to the similarity of their surface shapes. Nodes of tectogrammatical trees represent content words in sentences. [Haruno and Yamazaki, 1996] were engaged in alignment of content words only for Japanese-English pair, with the motivation similar to ours: it is not feasible to align functional words in structurally very different languages; however, they did not use any tree structure. Experiments with alignment of deep syntactic dependency trees are described for example in

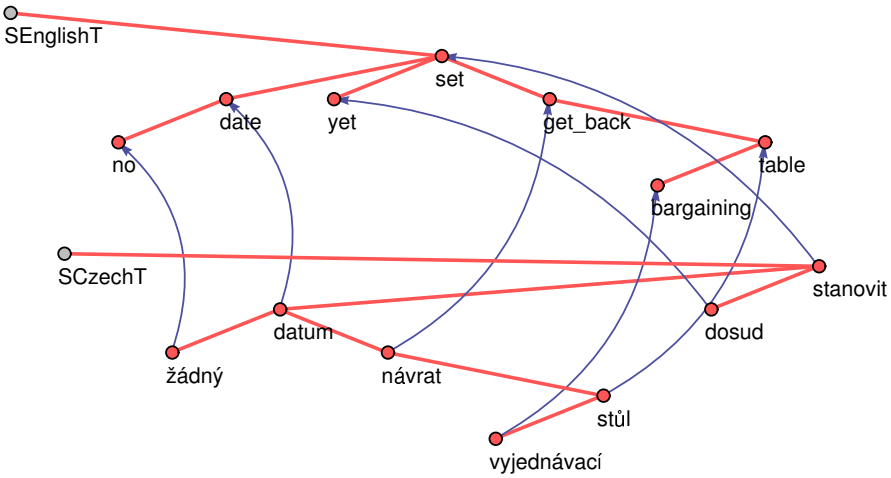


Figure 1.2: Example of alignment on the tectogrammatical layer. T-trees are simplified, only t-lemma attributes are depicted.

[Menezes and Richardson, 2001] and in [Bojar et al., 2007], but in our opinion no quantitative comparison of these approaches with our approach is possible due to different experiment contexts and goals.

Alignment on tectogrammatical layer for the same sentence as in Figure 1.1 is shown in Figure 1.2. 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 tectogrammatical trees are exactly those that were aligned as evident (full arrows) on the surface.

1.2 Goals of the Thesis

The goal of this thesis is to implement and evaluate a software tool for automatic alignment of Czech and English tectogrammatical trees (t-aligner for short). It will be implemented in TectoMT framework [Žabokrtský et al., 2008].

To evaluate the t-aligner it will be necessary to have manually aligned trees. We rejected the idea to manually align tectogrammatical trees. Trees are generated automatically and contain errors. The alignment of tectogrammatical trees will be shifted from sentences manually aligned on the word layer. It results more or less in choosing only the links between content words. The second goal of this thesis will be therefore to scheme up annotation rules for a word alignment, to lead the annotation process, and to evaluate it finally.

In the end we will compare the results of the t-aligner with other methods. The results will be also compared according to the types of text used for alignment.

1.3 Summary

In Chapter 2, we will describe some of the recent work concerning the alignment of trees. It comprises both the alignment of phrase structure trees and the alignment of dependency trees. There is also GIZA++ tool described. It is not really a tree aligner, but we can simply use it on linearized trees. After that follows the Section 2.3 giving information about resources of Czech-English parallel texts. The data samples from all of the resources were used for evaluation of our t-aligner.

The TectoMT framework, in which the t-aligner was implemented, is described in Chapter 3. There is also description of Prague Dependency Treebank annotation layers. Czech and English tectogrammatical analysis is depicted in the end of the chapter.

We give an account of the process of word alignment annotations in Chapter 4. The selection of data and types of connections are described here. Elementary annotation rules have been created throughout the duration of the first annotation. The tables with inter-annotator agreements follow. Finally, the transfer of word

alignment into the tectogrammatical layer is described. In this way we will acquire aligned tectogrammatical trees that can be used for t-aligner evaluation.

Chapter 5 concerns the implementation of the t-aligner. The alignment process is divided into two parts and described in detail. In Chapter 6 the evaluation methods are described. T-aligner is evaluated for the various types of data and its results are compared to the GIZA++ results. Chapter 7 contains conclusions and a discussion of the obtained results.

Chapter 2

Related Work

2.1 State of the Art in Tree-to-Tree Alignment

In this Section, we will describe some of already published algorithms for alignment of syntactic (or deeper syntactic) trees. The first two algorithms work with dependency trees that are very similar to tectogrammatical trees. They were tested on English-Spanish language pair and English-Japanese language pair respectively. The algorithm in Subsection 2.1.3 is different. We can get a word alignment by parsing both source and target sentences together. It is demonstrated on English-Chinese language pair. The last algorithm in Subsection 2.1.4 works with phrase trees.

2.1.1 A Best-First Alignment of Logical Forms

Arul Meneses and Stephen D. Richardson from Microsoft Research developed the tree-to-tree aligner of so-called Logical Forms of sentences. The Logical Form (LF for short) of a sentence is very similar to its tectogrammatical tree. It is an unordered graph representing the relations among the most meaningful elements of a sentence. Nodes are identified by the lemmas of the content words directed, labeled arcs indicate the underlying semantic relations. The Logical Form abstracts away from the surface word order, inflectional morphology, or functional words and it should have very similar structure for the same sentences in different languages. There is an example in Figure 2.1.

The alignment algorithm proceeds in two phases. In the first phase, it establishes tentative lexical correspondences between nodes in the source and target LFs. In the second phase, it aligns nodes based on these lexical correspondences as well as structural considerations. It starts from the nodes with the tightest lexical correspondence ("best first") and works outward from these anchor points.

To establish initial tentative word correspondences, a large bilingual dictionary together with the derivational morphology component is used. The algorithm also looks for matches between components of multi-word expressions and individual words. The tentative correspondences are depicted in Figure 2.1a.

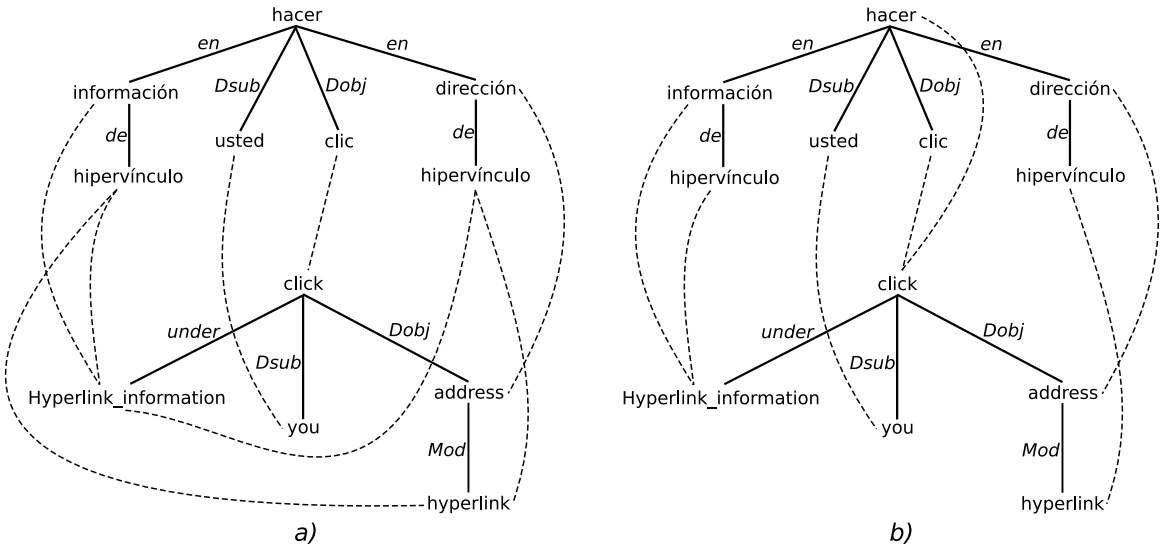


Figure 2.1: Logical Forms of Spanish-English pair: *En Información del hipervínculo, haga clic en la dirección del hipervínculo.* – *Under Hyperlink Information, click the hyperlink address.* a) lexical correspondences, b) alignment mappings, from [Menezes and Richardson, 2001]

The algorithm uses a set of alignment grammar rules that are ordered to create the most unambiguous alignments first and use these to disambiguate subsequent alignments. The algorithm is as follows:

- Initialize the set of unaligned source and target nodes to set of all source and target nodes respectively.
- Attempt to apply the alignment rules in the specified order, to each unaligned node or set of nodes in source and target. If the rule fails to apply to any unaligned node or set of nodes, move to the next rule.
- If all rules fail to apply to all nodes, exit. No more alignment is possible. (Some nodes may remain unaligned.)
- When a rule applies, mark the nodes or sets of nodes to which it applied as aligned to each other and remove them from the lists of unaligned source and target nodes respectively. Go to step 2 and apply rules again, starting from the first rule.

The alignment grammar includes the rules such as:

1. *Bidirectionally unique translation*: Align source node S with target node T if S has a lexical correspondence with T and with no other target node and T

has a lexical correspondence with S and with no other source node. Similarly for the set of nodes.

2. *Translation + Children*: Align S and T if S and T have a lexical correspondence and each child of S and T are already aligned to a child of the other.

3. *Translation + Parent*: Align S and T if S and T have a lexical correspondence and a parent node of S has already been aligned to a parent node of T.

4. *Verb + Object to Verb*: A verb V_1 (from either source or target) that has child C that is not a verb, but is already aligned to a verb V_2 and either V_2 has no unaligned parents, or V_1 and V_2 have children aligned to each other. Align V_1 and C to V_2.

5. *Parent + relationship*: Align nodes S and T if they have the same part-of-speech, no unaligned siblings, a parent P_S of S is already aligned to a parent P_T of T, and the relationship between P_S and S is the same as that between P_T and T.

Note that rules 4 and 5 rely solely on relationships between nodes. The alignment of the example trees is depicted in Figure 2.1b. In this case, the rules 1, 3, and 4 were used. For more detailed description of this algorithm, read the article [Menezes and Richardson, 2001].

2.1.2 Finding Word Correspondences in a Bilingual Parsed Corpus

Hideo Watanabe, Sadao Kurohashi, and Eiji Aramaki have very similar approach to alignment of dependency trees, see [Watanabe et al., 2003]. A dependency structure as they use is a tree consisting of nodes and arcs, where a node represents a content word and an arc represents a functional word or a relation between content words. For instance, as shown in Figure 2.2, a preposition *at* is represented as an arc in English.

The task is to find word correspondences between the nodes of a source tree and the nodes of a target tree. Word correspondences are found by consulting a bilingual dictionary. We denote word correspondence candidates by $WC(s,t)$, where s is a source node, and t is a target node. Most words can find a unique translation candidate in the target tree (this correspondence we denote by WA) but there are cases where more than one translation candidate exists in the target tree for a given source word.

Suppose a source word s has multiple candidate translation target words t_i, $i = 1, \ldots, n$. That is, there are multiple WCs originating form s. We denote them $WC(s,t_i)$. For each WC of s the procedure finds the neighbor WA correspondence whose distance to WC is below a threshold. The distance between $WC(s_1,t_1)$ and

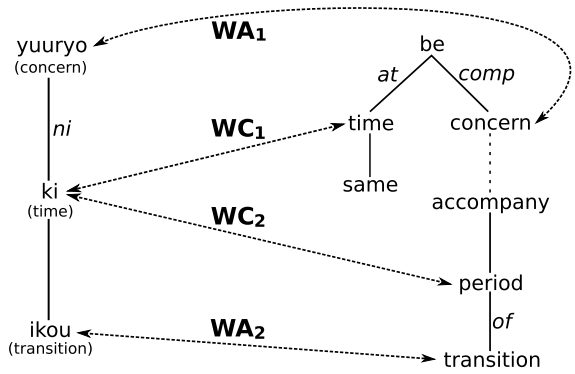


Figure 2.2: Example of word correspondences, from [Watanabe et al., 2003]

Input: $TreeS$, $TreeT$ – source and target dependency tree
Output: $wordcorrs$ – word correspondences

foreach $s \in TreeS$ **do**
    find the set of candidate translation nodes T;
    foreach $t \in T$ **do** make $WC(s,t)$ and add it to $wordcorrs$;
    if $|T| = 1$ **then** change $WC(s,t)$ to WA;
changed = **true**;
while *changed* **do**
    changed = **false**;
    foreach $s \in TreeS$ **do**
        $wps = \emptyset$;
        foreach WC originating from s in $wordcorrs$ **do**
            add its neighbor WA to wps;
        if $wps \neq \emptyset$ **then**
            find WC having the smallest distance to its neighbor WA in wps;
            change this WC to WX;
            delete all WCs whose source is s from $wordcorrs$;
            changed = **true**;
    foreach $W(s,t) \in wordcorrs$, $W(s,t)$ is not WC **do**
        if s has only 1 child s', which is a leaf
        and t has only 1 child t', which is a leaf **then**
            make $WS(s',t')$ and add it to $wordcorrs$;
            changed = **true**;
foreach $t \in TreeT$ **do**
    if there is only 1 WC which target is t **then** change WC to WZ;

Figure 2.3: Procedure for finding word correspondences

$WA(s_2, t_2)$ is defined as the distance between s_1 and s_2 plus the distance between t_1 and t_2), where a distance between two nodes is defined as the number of nodes in the path whose ends are the two nodes. Among WCs of s for which neighbor WA is found, the one with the smallest distance is chosen and other WCs are invalidated. We denote word correspondence found by this procedure as WX. The threshold value was set to 3. On the example in Figure 2.2, Japanese word *ki* has two English translation word candidates *time* and *period*. The correspondence WC_2 (the pair *ki* – *period*) wins because the distance between WC_2 and WA_2 is smaller than the distance between WC_1 and WA_1.

Correspondences $W(s, t)$, where s has only one child node which is a leaf and t ha also only one child node which is a leaf, are also considered. In this case, we construct a new word correspondence WS from these two leaf nodes. For instance, in Figure 2.2, if there is a word correspondence between *ki* and *period* and there is no word correspondence between *ikou* and transition, then $WS(ikou, transition)$ will be found by this step.

After applying the above WX and WS procedures, some target words t exist such that t is a destination of $WC(s, t)$ and there is no other WC whose destination is t. In this case, the $WC(s, t)$ correspondence candidate is chosen and is denoted as WZ word correspondence.

In Figure 2.3 there is a pseudo-algorithm of finding word correspondences.

2.1.3 Inversion Transduction Grammars

Dekai Wu describes in his article [Wu, 1997] Inversion Transduction Grammars (ITGs) which allow us to generate bilingual pairs of sentences. A simple transduction grammar is just a context-free grammar whose terminals are pairs of symbols. The notation e/ch is used for terminals to associate matching output tokens, where e is the English terminal and ch is the Chinese one. There is an example of simple transduction grammar:

S → [SP Stop]
SP → [NP VP] | [NP VV] | [NP V]
PP → [Prep NP]
NP → [Det NN] | [Det N]
NN → [A N] | [NN PP]
VP → [Aux VP] | [Aux VV] | [VV PP] | [PP VV]
VV → [V NP] | [Cop A]
Det → the/ε
Prep → to/向
N → authority/管理局 | secretary/司
A → accountable/负责 | financial/财政
Aux → will/将会
Cop → be/ε
Stop → ./。

There is a null symbol ϵ introduced and used in cases when the grammar generates a word only in one language. Terminal symbols ϵ/ch and e/ϵ are called singletons. If we use this grammar as classical context-free grammar and consider only first or only second part of terminals, we can simply generate the following pair of parse trees:

[[[The Authotity]$_{NP}$ [will [[be accountable]$_{VV}$ [to [the [[Financial Secretary]$_{NN}$]$_{NNN}$]$_{NP}$]$_{PP}$]$_{VP}$]$_{VP}$]$_{SP}$.]$_{S}$

[[[管理局]$_{NP}$ [将会 [[向 [[[财政 司]$_{NN}$]$_{NNN}$]$_{NP}$]$_{PP}$ [负责]$_{VV}$]$_{VP}$]$_{VP}$]$_{SP}$ 。]$_{S}$

A problem occurs, when we want to generate both sentences together. While the rule "VP → NP PP" was used in English, at the same place the inverse rule "VP → PP NP" was used in Chinese.

The order of the constituents in one language may be reverse in the other language for any given rule in ITG. The square brackets [ ] are used when the order is the same in both languages and angle brackets ⟨⟩ are used when the order is reversed. In given example we now replace the rule VP with this rule:

VP → [Aux VP] | [Aux VV] | ⟨VV PP⟩

With this Inversion Transduction Grammar we can already generate English sentence and its Chinese equivalent together:

[[[The/ε Authotity/管理局]$_{NP}$ [will/将会⟨[be/ε accountable/负责]$_{VV}$ [to/向 [the/ε [[Financial/财政 Secretary/司]$_{NN}$]$_{NNN}$]$_{NP}$]$_{PP}$ ⟩$_{VP}$]$_{VP}$]$_{SP}$ ./。]$_{S}$

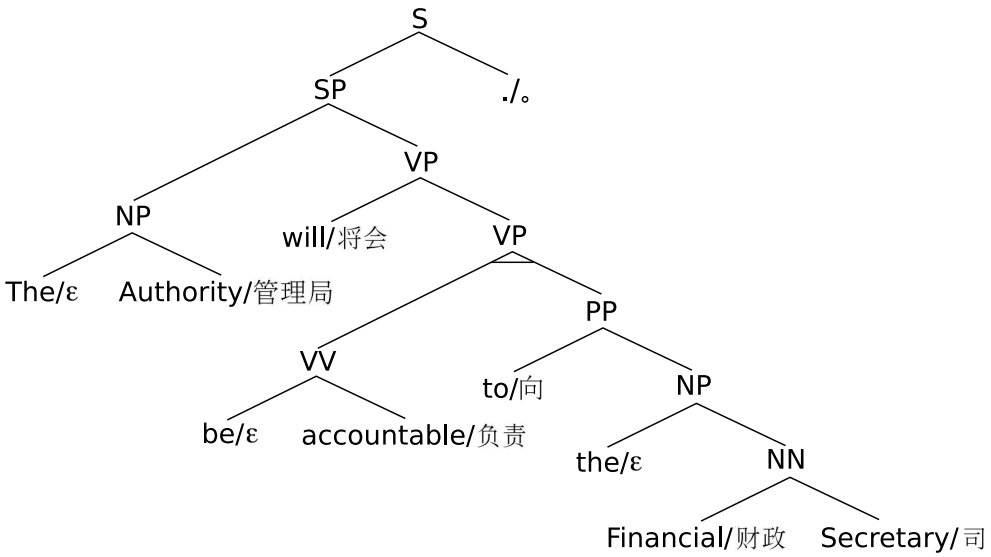


Figure 2.4: Inversion transduction parse tree, from [Wu, 1997]

Even though the order of constituents under the inner VP is inverted between the languages, an ITG can capture the common structure of the two sentences. This is compactly shown by writing the parse tree together for both sentences with the aid of an $\langle\rangle$ angle bracket notation marking parse tree nodes that instantiate rules of inverted orientation. In Figure 2.4 there is a parse tree example. The inversion at VP is illustrated by horizontal line.

In this case, alignments (phrasal or lexical) are a natural byproduct of bilingual parsing. Unlike "parse-parse-match" methods, this does not require a fancy grammar for both languages.

2.1.4 Language Pair-Independent Sub-Tree Alignment

John Tinsley, Ventsislav Zhechev, Mary Hearne and Andy Way present in their work [Tinsley et al., 2007] a robust aligner of phrase structure trees adhering to the following principles:

(i) independence with respect to language pair and constituent labelling schema
(ii) preservation of the given tree structures
(iii) minimal external resources required
(iv) word-level alignments not fixed a priori

A single external resource used are target-to-source and source-to-target word translation probabilities generated by running an automatic word aligner over the sentence pairs encoded in the bilingual treebank.

For a given tree pair (S, T), the alignment process is initialized by assigning scores $\gamma(s, t)$ to all hypothetical links (s, t) between nodes in S and T. All zero-scored links are blocked. The selection procedure then iteratively fixes on the highest-scoring link, blocking all hypotheses that contradict this link and the link itself, until no non-blocked hypotheses remain.

Given tree pair (S, T) and hypothetical link (s, t), the following strings are computed:

$$s_l = s_i \dots s_{ix} \qquad \bar{s_l} = S_1 \dots s_{i-1} s_{ix+1} \dots S_m$$
$$t_l = t_j \dots t_{jy} \qquad \bar{t_l} = T_1 \dots t_{j-1} t_{jy+1} \dots T_n,$$

where $s_i \dots s_{ix}$ and $t_j \dots t_{jy}$ denote the terminal sequences dominated by s and t respectively, and $S_1 \dots S_m$ and $T_1 \dots T_n$ denote the terminal sequences dominated by S and T respectively. There is an example in Figure 2.5. The dashed line denotes the link hypothesis. Then the scores are computed as follows:

$$\gamma(s, t) = \alpha(s_l, t_l) \cdot \alpha(t_l, s_l) \cdot \alpha(\bar{s_l}, \bar{t_l}) \cdot \alpha(\bar{t_l}, \bar{s_l})$$

Individual string-correspondence scores $\alpha(x, y)$ are computed using word alignment probabilities given by the Moses decoder [Koehn et al., 2007].

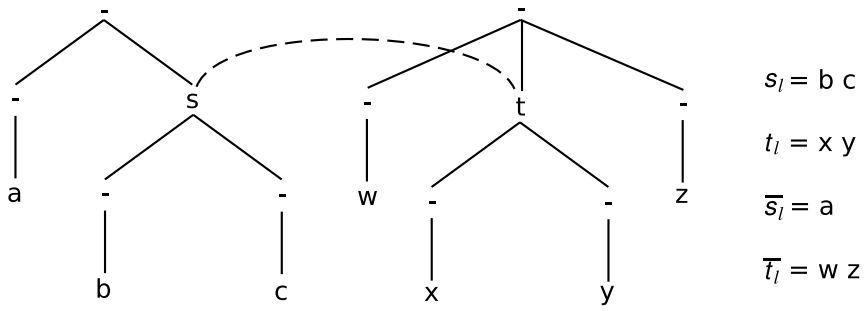


Figure 2.5: Strings computed for a given link hypothesis, from [Tinsley et al., 2007]

2.2 GIZA++ Alignment Tool

GIZA++ tool was designed by F. J. Och for word alignment of parallel corpora. It is an extension of the program GIZA, which was part of the Egypt system [Al-Onaizan et al., 1999], and supported by IBM Models 1, 2, and 3, as proposed in [Brown et al., 1993]. In GIZA++ there are available also models 4 and 5 (see [Och and Ney, 2000]). Brief description of IBM Models follows.

2.2.1 Alignment Models

IBM Model 1

Model 1 is the simplest model. It is based solely on lexical translation probability distributions. We define the translation probability for a Czech sentence $\overrightarrow{c} = (c_1, \ldots, c_{l_c})$ of length l_c to an English sentence $\overrightarrow{e} = (e_1, \ldots, e_{l_e})$ of length l_e with an alignment according to the function $a : j \rightarrow i$ as follows:

$$p(\overrightarrow{e}, a|\overrightarrow{c}) = \frac{\epsilon}{(l_c+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|c_{a(j)})$$

The parameter ϵ is the normalization constant, so that $p(\overrightarrow{e}, a|\overrightarrow{c})$ is a proper probability distribution.

There is a pseudo-code of EM training algorithm for Model 1 in the Figure 2.6. At the output we get a table of probabilities $t(e|c)$ for all possible Czech and English words. The probability $t(e|c)$ determines how likely we can translate the Czech word c with the English word e. Binding the word e to NULL means word-deletion, and binding the word c to NULL indicates word-insertion.

Input: *SentencePairs*, E – English dictionary, C – Czech dictionary
Output: table of translation probabilities $t(e|c)$

foreach (e, c), $e \in E$, $c \in C$ **do** $t(e|f) = 1$;
while not *convergence limit* **do**
  foreach (e, c), $e \in E$, $c \in C$ **do** $count(e|c) = 0$;
  foreach $c \in C$ **do** $total(c) = 0$;
  foreach $(e\_s, c\_s) \in SentencePairs$ **do**
    foreach $e \in e\_s$ **do**
      $total_s(e) = 0$;
      foreach $c \in c\_s$ **do** $total\_s(e)$ += $t(e|f)$;
    foreach $e \in e\_s$ **do**
      foreach $c \in c\_s$ **do**
        $count(e|c)$ += $t(e|c)$ / $total\_s(e)$;
        $total(c)$ += $t(e|c)$ / $total\_s(e)$;
  foreach $c \in C$ **do**
    foreach $e \in E$ **do**
      $t(e|c) = count(e|c)$ / $total(c)$;

Figure 2.6: EM training algorithm for IBM Model 1

IBM Model 2

An explicit model for alignment is added in IBM Model 2. The translation of a Czech input word in position i to an English word in position j is modeled by an alignment probability distribution $a(i|j, l_e, l_c)$. We can view translation under IBM Model 2 as a two step process process with a lexical translation step (IBM Model 1) and an alignment step. The two steps are combined mathematically as:

$$p(\overrightarrow{e}, a|\overrightarrow{c}) = \epsilon \prod_{j=1}^{l_e} t(e_j|c_{a(j)}) a(a(j)|j, l_e, l_c)$$

IBM Model 3

Model 3 introduces word fertility table $n(\phi|c)$, which indicates the probability of the number of foreign words induced from a given Czech word. For example in Figure 2.2.1, the Czech word "nekupuji" induces two English words "not" and "buy", while the Czech word "si" induces no English word – its fertility is 0.

After the fertility step the NULL insertion step comes. The NULL tokens are inserted for target words that have no counterpart in the source sentence. For example, the English word "do" is often inserted when translating verbal negations. The third step is lexical translation as in Model 1. Finally, the distortion is modeled almost the same way as in Model 2 with a probability distribution $d(j|i, l_e, l_c)$.

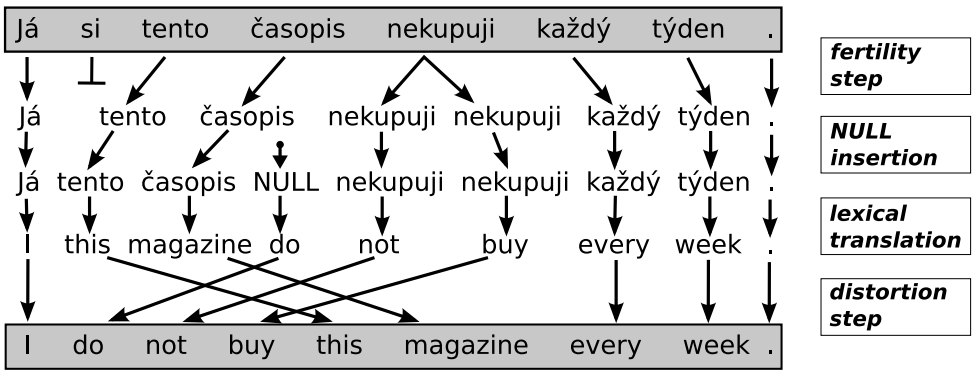


Figure 2.7: IBM Model 3

IBM Model 4

Model 4 comes with more intuitive handling of distortion than the preceding models, where word reordering depended only on the length of the sentences, completely ignoring the words in both languages. Model 4 deals with word classes and relative positioning. Word classes ($C(e)$, $C(c)$) are automatically derived from both languages independently using a clustering algorithm [Brown et al., 1992]. For this model the relative distortion model is introduced. The placement of the translation of an input word is typically based on the placement of the translation of the proceeding input word.

2.2.2 Symmetrization Methods

The output form GIZA++ is asymmetric, because at most one counterpart in the target language is found for each word in the source language. To symmetrize the alignment, we run GIZA++ in both directions (source-to-target and also target-to-source) and get two different alignments. There is an example of the two outputs in Figure 2.2.2. To establish word-alignment based on the two GIZA++ alignments, several heuristics may be applied. The most widely used are intersection and grow-diag-final methods. There is a pseudo-code describing all the methods in Figure 2.9.

- **srctotgt:** We only consider word-to-word alignments from the source-target GIZA++ alignment file.
- **tgttosrc:** We only consider word-to-word alignments from the target-source GIZA++ alignment file.
- **union:** The union of the two GIZA++ alignments is taken. All word alignment points that occur at least in one alignment are preserved. See Figure 2.2.2a and procedure `Union()` in pseudo-code 2.9.

- **intersection:** The intersection of the two GIZA++ alignments is taken. Only word alignment points that occur in both alignments are preserved. See Figure 2.2.2b and procedure `Intersection()` in pseudo-code 2.9.
- **grow:** At first, the **intersection** of the two GIZA++ alignments is made. In the growing step, additional alignment points are added. Only such alignment points that are in the **union** are considered. Potential alignment points that neighbor with already established alignment points are added. In this case, the neighborhood is defined as directly left, right, top, and bottom point. See the procedure `Grow()` in pseudo-code 2.9. The grow step is marked with "G" in the Figure 2.2.2c.
- **grow-diag:** Similarly as the **grow** symmetrization. Only the neighborhood also includes other four points, which neighbor diagonally. See the `GrowDiag()` procedure in pseudo-code 2.9. The grow-diag step is marked with "G" and "GD" in Figure 2.2.2c.
- **grow-diag-final:** At first, the **grow-diag** symmetrization is done. In a final step, alignment points that were in one of the GIZA++ alignments and do not neighbor with established alignment points are added. It is done for alignment points between words, where at least one of them is currently unaligned. See the procedure `GrowDiagFinal()` in pseudo-code 2.9. The final step is marked with "F" in Figure 2.2.2c.
- **grow-diag-final-and:** Similarly as for **grow-diag-final** but only alignment points that are between two unaligned words are added. See the procedure `GrowDiagFinalAnd()` in pseudo-code 2.9.

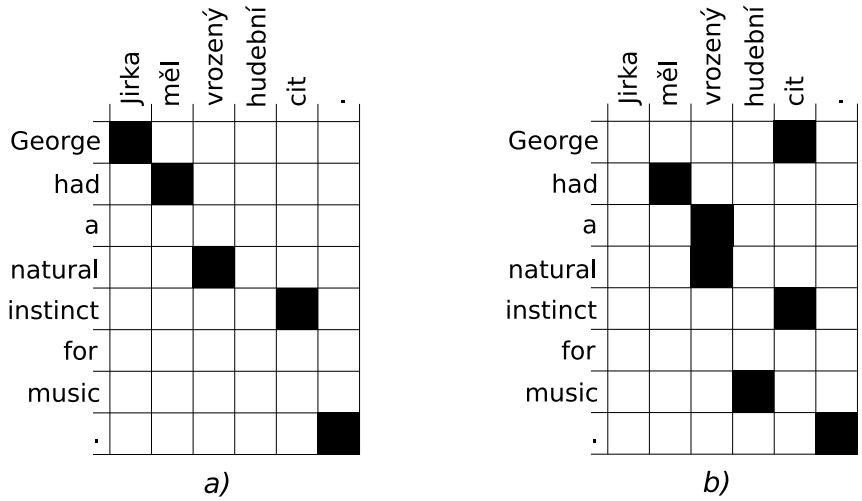


Figure 2.8: Two GIZA++ outputs: a) source-target, b) target-source

```
procedure DoGrowing(e2f, f2e, neighbors);
  new_points_added = true;
  while new_points_added do
    new_points_added = false;
    foreach (e, f) ∈ (0 ... en, 0 ... fn) do
      if Aligned(e, f) then
        foreach (et, ft) ∈ neighbors do
          new_e = e + et;
          new_f = f + ft;
          if not IsAligned(e_new) and not IsAligned(f_new)
            and (e_new, f_new) ∈ e2f ∪ f2e then
            Align(e_new, f_new);
            new_points_added = true;
procedure Final(a);
  foreach (e, f) ∈ (0 ... en, 0 ... fn) do
    if (not IsAligned(e) or not IsAligned(f)) and (e, f) ∈ a then
      Align(e, f);
procedure FinalAnd(a);
  foreach (e, f) ∈ (0 ... en, 0 ... fn) do
    if not IsAligned(e) and not IsAligned(f) and (e, f) ∈ a then
      Align(e, f);
procedure Intersection(e2f, f2e);
  foreach (e, f) ∈ e2f ∩ f2e do Align(e, f);
procedure Union(e2f, f2e);
  foreach (e, f) ∈ e2f ∪ f2e do Align(e, f);
procedure Grow(e2f, f2e);
  Intersection(e2f, f2e);
  neighbors = ((-1,0),(1,0),(0,-1),(0,1));
  DoGrowing(e2f, f2e, neighbors);
procedure GrowDiag(e2f, f2e);
  Intersection(e2f, f2e);
  neighbors = ((-1,-1),(-1,0),(-1,1),(0,-1),(0,1),(1,-1),(1,0),(1,1));
  DoGrowing(e2f, f2e, neighbors);
procedure GrowDiagFinal(e2f, f2e);
  GrowDiag(e2f, f2e);
  Final(e2f);
  Final(f2e);
procedure GrowDiagFinalAnd(e2f, f2e);
  GrowDiag(e2f, f2e);
  FinalAnd(e2f);
  FinalAnd(f2e);
```

Figure 2.9: Symmetrization methods in pseudo-code

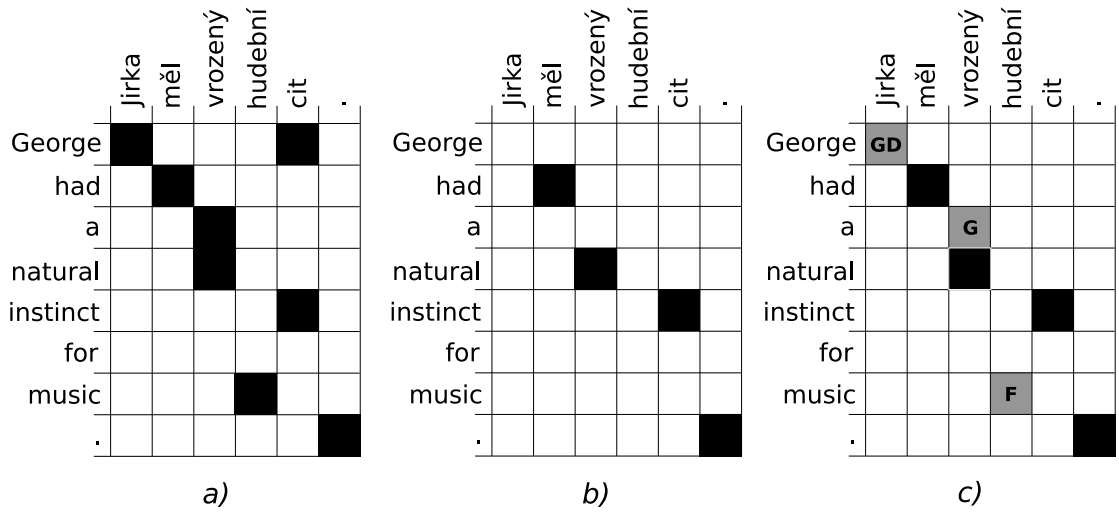


Figure 2.10: Symmetrization methods: a) union, b) intersection, c) grow-diag-final

2.3 Resources of Parallel Texts for Czech and English

Parallel corpora are used for comparative language study. In computational linguistics, the statistical analysis can be used to discover patterns between languages, with little or no linguistic information. The description of parallel corpora including the Czech-English pair follows.

2.3.1 Acquis Communautaire Parallel Corpus

The Acquis Communautaire [Ralf et al., 2006] is the total body of European Union law applicable to the EU Member States. This collection of legislative text changes continuously and currently comprises selected texts written between the 1950s and now. The Acquis Communautaire texts exist in the following 22 languages: Bulgarian, Czech, Danish, German, Greek, English, Spanish, Estonian, Finnish, French, Hungarian, Italian, Lithuanian, Latvian, Maltese, Dutch, Polish, Portuguese, Romanian, Slovak, Slovene and Swedish.

The corpus contains about 460,000 texts and a total of over one billion words. There is more than 20,000 documents translated into all 22 languages. Strictly speaking, the corpus is currently aligned at the paragraph level. However, the paragraphs of the corpus are usually short and do usually contain one sentence, or even only part of a sentence.

2.3.2 Kačenka

The parallel corpus KAČENKA (Korpus Anglicko-Český – Elektronický Nástroj Katedry Anglistiky) has been created by the Department of English, Faculty of Arts, Masaryk University during the year 1997 to support research and teaching in the field of translation. See [Rambousek et al., 1997] for details.

The idea of the authors was to create a small parallel corpus which would enable to work with entire texts in translation analysis rather then short extracts. It contains 30 books and 2 other non-literary texts translated from English to Czech and it makes more than 3,000,000 words. Roughly one half of this corpus have been acquired by means of scanning. The texts are aligned on the sentence level.

2.3.3 Prague Czech-English Dependency Treebank

Prague Czech English Dependency Treebank (PCEDT, see [Cuřín et al., 2004] for details) is a corpus of Czech-English parallel resources suitable for experiments in machine translation, with a special emphasis on dependency-based (structural) translation (with evaluation data provided for Czech-to-English systems). The core part is a Czech translation of 21,600 English sentences from the Wall Street Journal part of Penn Treebank corpus.

PCEDT (version 1.0) contains more than 21,000 sentence pairs (about one million Czech and English words). Sentences of the Czech translation were automatically morphologically annotated and parsed into analytical and tectogrammatical level, according to the Prague Dependency Treebank schema (see [Hajič et al., 2006]). The original English sentences were transformed from the Penn Treebank phrase-structure trees into dependency representations. A held-out (development and evaluation) set of 515 sentence pairs was selected and manually annotated on tectogrammatical level in both Czech and English; for the purposes of quantitative evaluation this set has been retranslated from Czech to English by 4 different translation companies.

PCEDT also comprises a parallel Czech-English corpus of plain text from Reader's Digest 1993-1996 consisting of 53,000 parallel sentences.

2.3.4 CzEng

The Czech-English parallel corpus CzEng (see [Bojar and Žabokrtský, 2006] for details) consists of a large set of parallel texts form the publicly available sources in an electronic form. The main purpose of the corpus is to support Czech-English and English-Czech machine translation research. It also contains parts of corpora described herein before.

In the current version 0.7, the majority of the data are the Czech and English documents from *Acquis Communautaire* corpus. There is also translated *EU con-*

stitution, stories form *Reader's digest*, articles from *Project Syndicate*, *KDE* and *GNOME* localization files, anonymous user translations (*Navajo*), and literary texts (5 books form the corpus *Kačenka* and other 5 *E-books* available freely on the Internet).

Chapter 3

TectoMT Framework

The tectogrammatical MT system, see [Žabokrtský et al., 2008], was primarily build for a high-quality linguistically motivated translation using the Prague Dependency Treebank layered framework (PDT, see [Hajič et al., 2006]). It is also useful for testing the true usefulness of various NLP tools within a real-life application.

TectoMT is written in Perl and is based on technologies from PDT 2.0 such as `tred/btred/ntred` and PML. Special attention is paid to modularity: We can decompose the task into a sequence of processing modules (called blocks) with relatively tiny, well-defined sub-tasks, so that each module is independently testable, improvable, or substitutable.

There are modules for analyses, transfer, syntheses, alignment, and evaluation. We can easily swap the modules or make new chains of modules for solving the tasks. All modules works with the same XML based data format. We can view any stage of our task in the TrEd application.

3.1 Prague Dependency Treebank

In the TectoMT system we use the layers of language description defined in the Prague Dependency Treebank 2.0 (PDT) described in [Hajič et al., 2006]. It is based on the Functional Generative Description, developed by Petr Sgall and his collaborators since 1960s (see [Sgall, 1967]) and 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).

3.1.1 Morphological Layer

On the morphological layer, the sentence consists of a sequence of tokens. Each token corresponds either to one word or to non-alpha-numerical character (e.g. punctuation, other symbols) and has three attributes: word form, morphological lemma and tag.

Since Czech is a language with rich inflection, the tagset used is very large. There are about 1100 tags in PDT out of 4257 theoretically possible. But most of the tags are used very rarely. The tag consists of 15 characters, each position represents one morphological category: Part of speech, Detailed part of speech, Gender, Number, Case, Possessor's gender, Possessor's number, Person, Tense, Voice, Degree of comparison, Negation, two reserve positions, and Variant. Complete description of the morphological annotation can be found in [Hana et al., 2005].

For English, we use Penn Treebank POS annotation [Marcus et al., 1993]. This annotation uses only 48 tags.

3.1.2 Analytical Layer

On the analytical layer, a rooted dependency tree is being build for every sentence. Every token from the morphological layer becomes a node in the analytical tree. Only one node – the "technical" root – is added. The analytical function is assigned to each node. In fact, it is the type of dependency relation between the node and its parent node.

Coordinations and appositions are technically also handled by "dependency" labels. The appropriate conjunction is the parent node and the coordination members are its children. They are marked as coordinated structure members, so that we can distinguish them from their common modifiers that also depends on the coordinating conjunction.

Each node has one of 28 analytical functions, such as: `Pred` (predicate), `Sb` (subject), `Obj` (object), `Adv` (adverbial), `Atv` (complement), `Atr` (attribute), `Pnom` (nominal predicate), `AuxV` (auxiliary verb "be"), `Coord` (coordination node), `AuxP` (preposition), `AuxC` (subordinating conjunction), `AuxS` (root of the tree), `ExD` (technical value for ellipsis), etc. See [Hajičová et al., 1999] for details.

3.1.3 Tectogrammatical Layer

On the tectogrammatical layer there are also dependency trees but unlike the analytical layer, only auto-semantic words have their own nodes here. Function words like auxiliary verbs, subordinating conjunctions, or prepositions are represented in the respective nodes in the form of their attributes.

The tectogrammatical nodes (t-nodes for short) are linearly ordered according to their increasing communicative dynamism (the `deepord` attribute). For each t-node the contextually bounded children are always before the contextually unbounded ones.

There are two types of links from t-nodes to their corresponding nodes in analytical trees. The `lex.rf` attribute is referencing to the appropriate "auto-semantic" a-node, while the `aux.rf` attribute is referencing to the corresponding auxiliary a-nodes that have not their own t-nodes. Ellipsis (surface-deleted nodes) are added.

Some of the other attributes of t-nodes follow: Each t-node has a tectogrammatical lemma (`t_lemma`). `Functor` determines the type of semantic relation between the t-node and its parent. `Sempos` is the semantic part of speech. Grammatemes comprise a group of attributes that are the semantically-oriented counterparts of morphological categories such as aspect, degree of comparison, modality, gender, iterativeness, negation, number, person, and tense.

Further description of the Czech tectogrammatical annotation scheme can be found in [Böhmová et al., 2005]. The annotation scheme for English was described in [Cinková et al., 2006].

Tectogrammatical trees are slightly simplified in TectoMT. There are no "copied" t-nodes and the linear t-node order corresponds to the word order.

3.2 Tectogrammatical Machine Translation

Vauquois MT triangle in Figure 3.1 shows the procedure of translation via tectogrammatical layer. The source text is first analyzed (see Section 3.3). Produced source language tectogrammatical trees are then transfered into the target language tectogrammatical trees and from these trees the target text is generated finally. You can find more detailed description in [Bojar et al., 2007] and in [Žabokrtský et al., 2008].

The idea of using tectogrammatics as the transfer layer has advantages and disadvantages. It is sufficiently abstract in point of inflection and functional words.

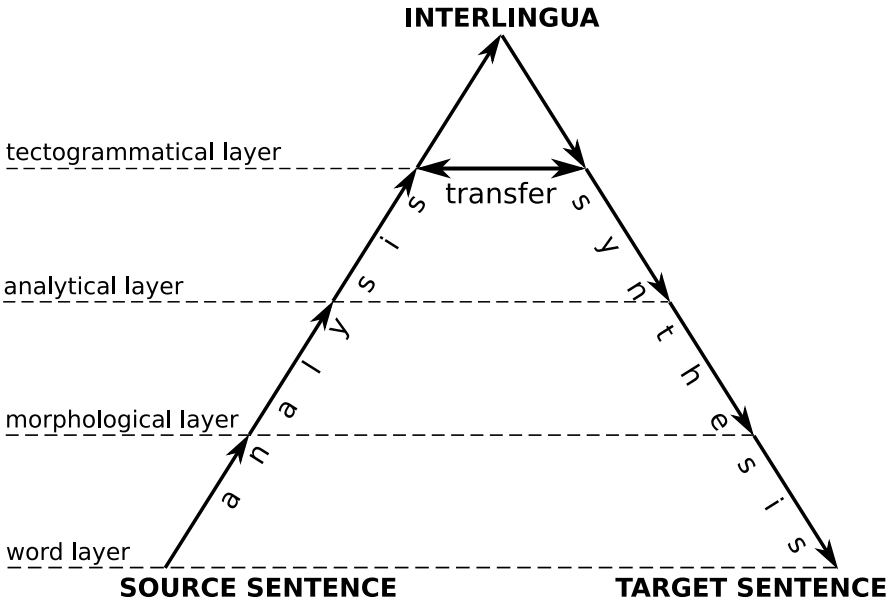


Figure 3.1: Vauquois MT triangle in terms of PDT

T-nodes correspond to autosemantic words only. Tectogrammatical trees are more similar and therefore fewer structural changes are needed in the transfer step. Local tree contexts in trees also carry more information than local linear contexts in the original sentences.

Big disadvantage of the tectogrammatical machine translation is the fact that many mistakes occur during analysis and generation phases.

3.3 Czech and English Tectogrammatical Analysis

For the alignment of Czech and English tectogrammatical trees, the tectogrammatical analysis of both source and target language is required. In this section we will list the tools and show examples of Czech and English analysis.

Czech sentences are first tokenized, morphologically analyzed, and disambiguated by the morphological tagger shipped with PDT 2.0 [Hajič et al., 2006]. One example is in Figure 3.2. Next comes the syntactic analysis realized by McDonald's MST parser [McDonald et al., 2005]. The analytical trees are then automatically converted into tectogrammatical trees. Analytical and tectogrammatical trees are shown in Figures 3.3 and 3.4 respectively.

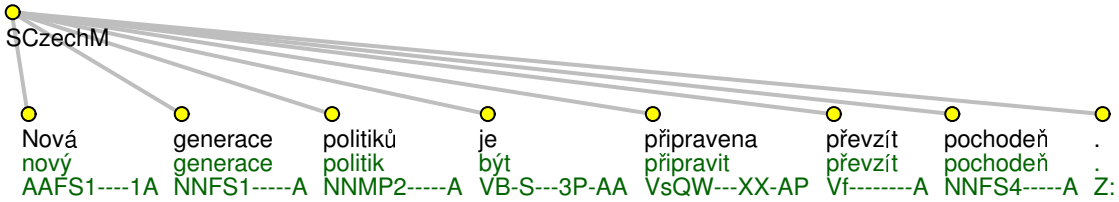


Figure 3.2: Czech morphological layer

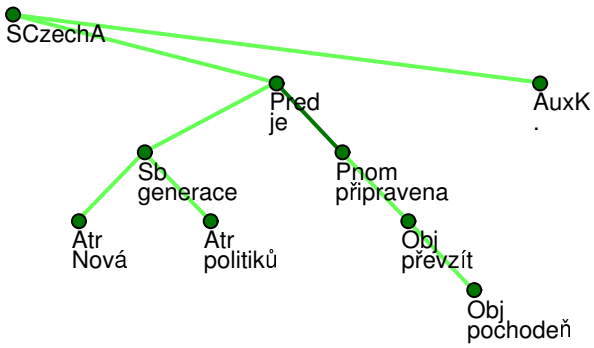


Figure 3.3: Czech analytical tree

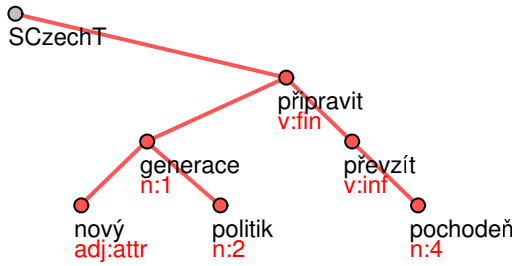


Figure 3.4: Czech tectogrammatical tree

English sentences are tokenized and tagged by the TnT tagger [Brants, 2000], see example in Figure 3.5. Then they are syntactically analyzed by the Collins parser [Collins, 1999]. Phrase trees (Figure 3.6) are converted into dependencies (Figure 3.7) and finally into the tectogrammatical trees (Figure 3.8).

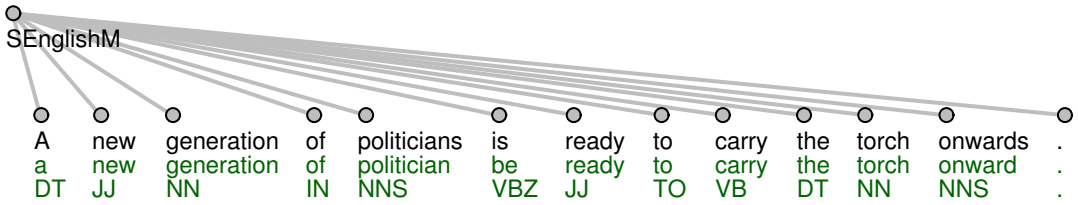


Figure 3.5: English morphological layer

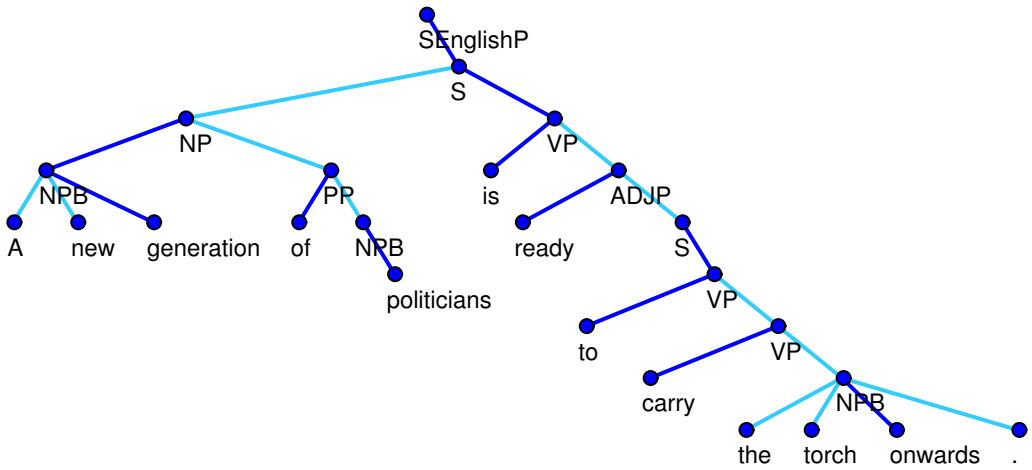


Figure 3.6: English phrase tree

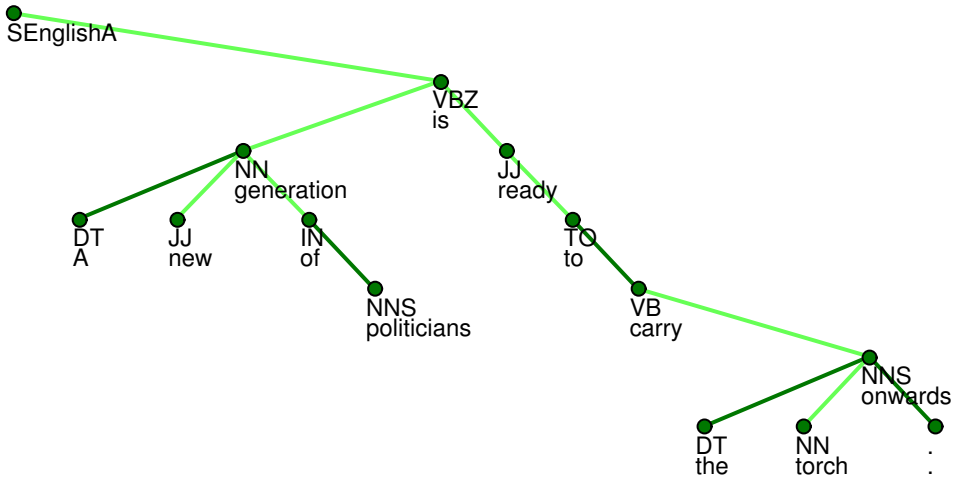


Figure 3.7: English analytical tree

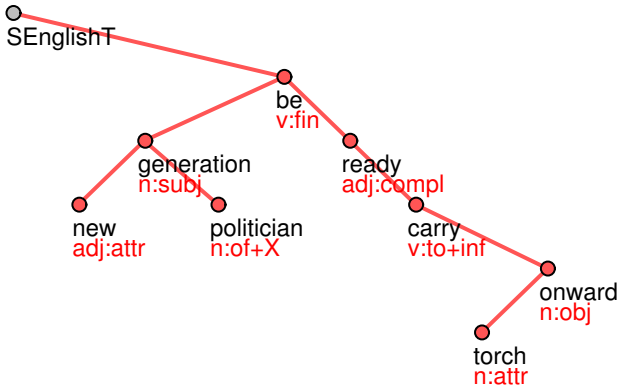


Figure 3.8: English tectogrammatical tree

Chapter 4

Manual Word-Alignment

The gold standard – manually aligned data – allow us to measure the accuracy of automatic aligners. For our purpose we should align manually a set of tectogrammatical tree pairs. But this is not feasible. One reason is that the trained aligners would be then less robust on automatically generated trees. The sentences would have to be also analyzed manually and it would take a lot of time. Second reason is the flexibility. Any changes in t-tree scheme would involve complete check-up of the trees and eventually re-aligning.

We decided to align sentences on the word level. The word alignment can be simply transformed into the tectogrammatical one using the `lex.rf` links. We exclude the alignment links from/to the tokens that do not have their own tectogrammatical nodes.

The only preprocessing before the word alignment is tokenization. If we already have manual aligned sentences and we would change the tokenization, we could simply re-align them automatically using several rules.

4.1 Data Selection and Preprocessing

We used the data form the corpus CzEng, version 0.7. We decided to choose samples of all types of sources. We did not use *KDE* and *GNOME* localization files and *Navajo User Translations* because this data are not really sentences, there are mainly individual phrases or words. We selected about 500 sentence pairs form EU laws, 500 pairs from *Project Syndicate* and 500 pairs from books and *Reader's Digest*. In Table 4.1 there are the properties of the data chosen for manual word alignment. It contains also the development and evaluation data from Prague Czech-English Dependency Treebank (*PCEDT*, see [Cuřín et al., 2004] for details), which were already aligned before (see [Bojar and Prokopová, 2006]).

From the selected documents we copied chunks of roughly 50 sentence pairs. Sometimes the sentences on Czech and English side did not match exactly (there were not only 1:1 relations). In this case we either split the sentence in one language or join several sentences in the other language in order to have only 1:1 relations at the output.

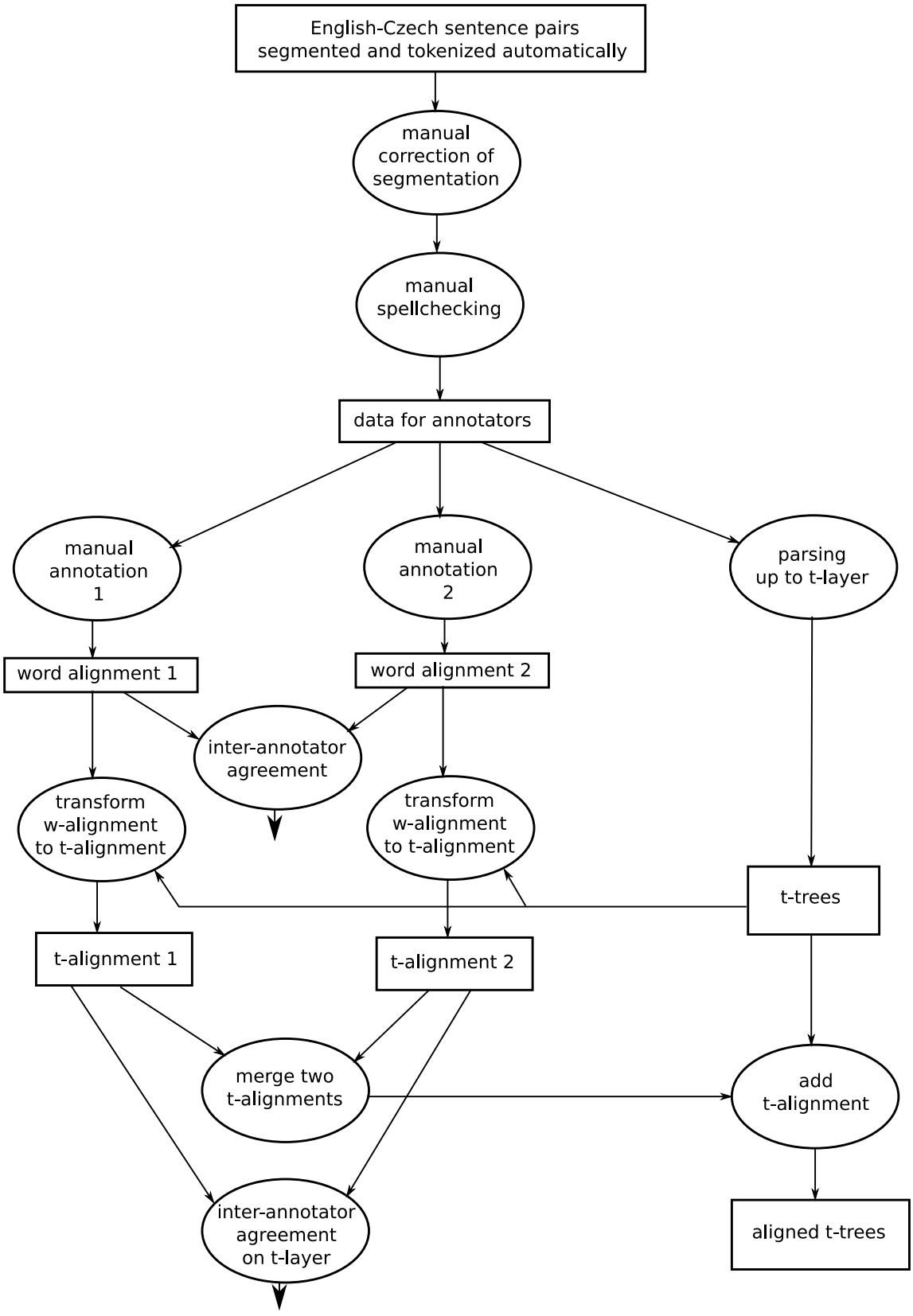


Figure 4.1: Data flow diagram of the manual word alignment process

Table 4.1: Data chosen from CzEng and PCEDT

source	chunks	sentences	EN tokens	CS tokens	all tokens
Acquis Communautaire	10	501	13,512	10,752	24,264
Reader's Digest	7	350	6,294	5,792	12,086
Project Syndicate	10	484	10,714	9,990	20,704
Kačenka	2	100	3,006	2,553	5,559
E-Books	1	50	797	633	1,430
P. Synd. (Named Entities)	168	500	12,799	11,052	23,851
PCEDT	22	515	12,697	12,174	24,871
Total	**190**	**2500**	**59,819**	**52,946**	**112,765**

There were also extracted other 500 sentence pairs from *Project Syndicate*. It was sentence pairs in 1:1 relation only, in which there was a relatively high presence of named entities (names of persons, countries, corporations etc.). This data are in the chunks of only about three sentences and not intersect the previous data from *Project Syndicate*. We will call them "Project Syndicate (Named Entities)".

There is the data flow diagram of the manual word alignment process in Figure 4.1. All the English and Czech sentences were converted to the same format and tokenized. Slightly modified Penn Treebank style tokenization [Marcus et al., 1993] was used for English. Czech tokenizer is very simple – each non-alphanumeric and non-white character is an extra token and all alphanumeric sequences (words) are tokens. After the correction of segmentation, manual spell-checking was done. The sentences were then given to two annotators to align it.

4.2 Alignment Types and Rules

The task for annotators is to mark links between Czech and English tokens, which corresponds to each other. No, one or more links can lead from/to each token. Following [Bojar and Prokopová, 2006] we used three types of links:

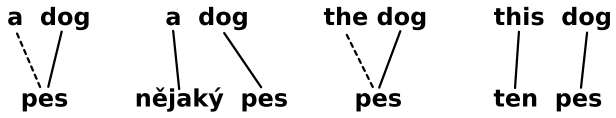
- **SURE link** – The individual words match.
- **PHRASAL link** – Whole phrases correspond but not words by themselves. We link each word in the Czech phrase to every word in the English phrase.
- **POSSIBLE link** – The connection is possible though doubtful. This type of link is used especially to connect words that do not have a real equivalent in the other language but syntactically clearly belong to a word nearby, such as English articles.

For phrasal alignments, annotators were encouraged to align also individual words in the phrases using sure or possible alignments, if reasonable. They were also instructed to use phrasal links as less as possible.

It is clear that this description of the task is not sufficient. We have to declare how to align common language constructions. It concerns mainly the functional words. The recommendations how to deal with the possible links follow:

4.2.1 Articles

If an English article corresponds to a Czech demonstrative or indefinite pronoun (e. g. *ten*, *nějaký*, ...) we link them together. In other cases we link the article by possible link to the appropriate Czech noun.



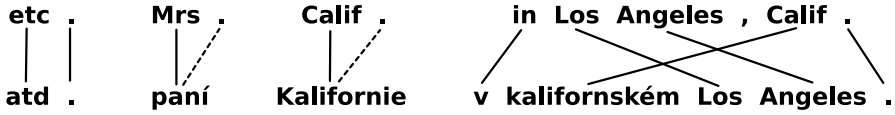
4.2.2 Prepositions

If two prepositions correspond to each other we link them by sure link. We do this even if the prepositions have not generally the same meaning. If a preposition occurs only in one sentence, we link it by possible link to an appropriate noun in the other sentence.

4.2.3 Punctuation

We link together two commas that occur in both sentences in the same position. If the comma is only in one sentence and there is a conjunction in the other sentence, we link the comma to this conjunction by possible link.

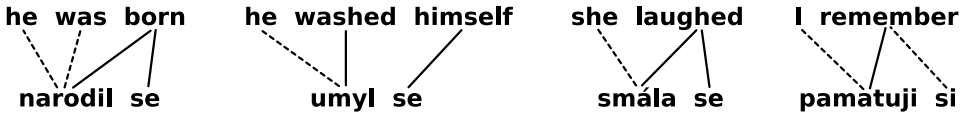
Where two abbreviations correspond, we link them by sure link as well as the following dots. If the full word corresponds to an abbreviation, we link its dot by possible link to the full word. If the abbreviation is at the end of a sentence before full-stop, we classify it as the abbreviation without a dot.



4.2.4 Pronouns

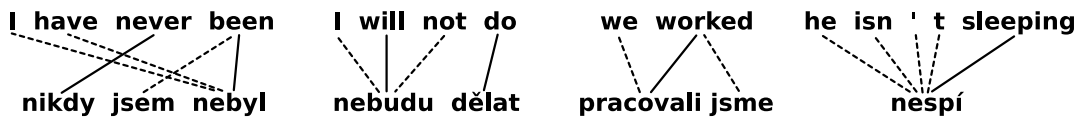
If an English personal pronoun does not have its own counterpart in Czech sentence we link it by possible link to the finite Czech verb. In case a pronoun is used in one language but in the other language there is a noun as its counterpart, we do not link them. If an English possessive pronoun does not have its counterpart in Czech but it is obliged here, we link it by possible link to an appropriate Czech noun, otherwise we do not link it.

We link the Czech reflexive pronouns (*si*, *se*) to their counterparts (e. g. *myself*, *yourself*, ...). If it has no counterpart, we link it to the appropriate verb by possible link. In case it is reflexivum tantum, we use the sure link.



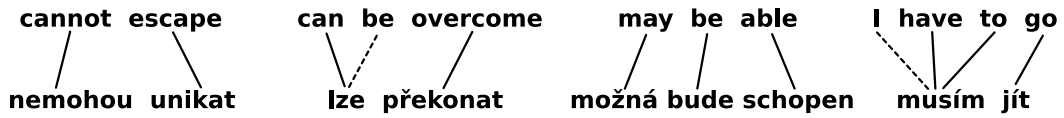
4.2.5 Auxiliary Verbs

If an English auxiliary verb (*be*, *do*, *have*) does not have its counterpart with the same meaning in Czech, we link it to the Czech finite verb with possible link. Similarly for the Czech auxiliary verb *být*. We never link auxiliary verbs to personal pronouns.



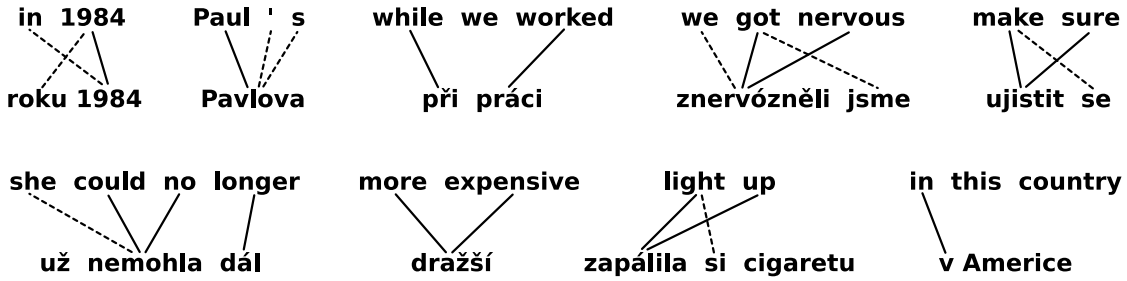
4.2.6 Modal Verbs

If a modal verb occurs only in one language, we do not link it. We also link possible personal pronouns and auxiliary verbs to the modal verb.



4.2.7 Miscellaneous

The basic rules that were introduced above can not cover all possible phenomena at all. All the remaining cases depended on consideration of annotators.



4.3 Inter-Annotator Agreement

Inter-annotator agreement (IAA for short) shows us the reliability of manual annotation. It measures a similarity of the two independent annotations. We compute it as F-measure on data of one annotator, while the data of the other are virtually treated as gold standard.

We can define $p_{A_1A_2}$ and $p_{A_2A_1}$ as precision of the annotator A_1 in reference to the annotator A_2 and reversely:

$$p_{A_1A_2} = \frac{|A_1 \cap A_2|}{|A_1|}, \qquad p_{A_2A_1} = \frac{|A_1 \cap A_2|}{|A_2|},$$

where $|A_1 \cap A_2|$ denotes a number of links that were made by both annotators. If we want to distinguish the types of links we count into $|A_1 \cap A_2|$ only links of the same type in both annotations. We compute IAA as the harmonic mean of the two mutual precisions.

$$IAA(A_1, A_2) = \frac{2 \cdot p_{A_1A_2} \cdot p_{A_2A_1}}{p_{A_1A_2} + p_{A_2A_1}} = \frac{2 \cdot |A_1 \cap A_2|}{|A_1| + |A_2|}$$

We have 2500 manually aligned Czech-English sentence pairs for the evaluation. The data were split into the 5 groups according to their type in the following way:

1. **Acquis Communautaire** – 501 sentences from EU laws
2. **Project Syndicate** – articles, 484 sentences
3. **Reader's Digest, Kačenka, Books** – literary texts from CzEng. This group includes seven stories from Reader's Digest and parts of three books – Charles Dickens/Oliver Twist, Thomas Hardy/Tess of the d'Urbervilles, and Jerome K. Jerome/Three Men in a Boat. (500 sentences)
4. **PCEDT** – already annotated 515 sentences, see [Bojar and Prokopová, 2006]
5. **Project Syndicate (Named Entities)** – 500 sentences that contain occurrences of named entities.

Counts of connections made by two annotators A1 and A2 are in Table 4.2. We can see that the biggest difference was in the category of phrasal links. The reason follows: The decision, whether to connect Czech and English phrases by phrasal links or to use several sure and possible links and some words leave without connection, is problematic. Each annotator feel it a bit differently and each one has the boundary somewhere else. The difference between the counts of phrasal links used is so great also because the annotator who decided to use phrasal links makes many links at once. (It is necessary to connect all words in the Czech phrase to all words in the English phrase.)

In Figure 4.3 there are statistics of annotator agreement and disagreement. Each column denotes one possible combination of two types of link. For example, the column *sure – possible* shows how many links has been labeled by one annotator as *sure* and by the other annotator as *possible* (or conversely). Besides the absolute

Table 4.2: Manual word-alignment statistics

Data source	Sent.	Sure		Possible		Phrasal	
		A1	A2	A1	A2	A1	A2
Acquis Communautaire	501	9165	9637	3662	3622	366	213
Project Syndicate	484	7335	8135	2809	2747	875	305
Reader's Digest, Kačenka, Books	500	6265	6866	2638	3093	1240	820
PCEDT	515	10784	11009	1831	1895	1936	580
Proj. Synd. (Named Entities)	500	9559	9623	2246	2949	209	473
Total	2500	43108	45270	13186	14306	4696	2391

Table 4.3: Occurrences of annotator agreement and disagreement

sure – sure	possible – possible	phrasal – phrasal	sure – possible	sure – phrasal	sure – no link	possible – phrasal	possible – no link	phrasal – no link
Acquis Communautaire								
8,835	2,655	69	533	127	472	57	1,384	257
61.4%	18.5%	0.5%	3.7%	0.9%	3.3%	0.4%	9.6%	1.8%
Project Syndicate								
7,116	1,657	195	507	152	579	118	1,617	520
57.1%	13.3%	1.6%	4.1%	1.2%	4.6%	0.9%	12.9%	4.2%
Reader's Digest, Kačenka, Books								
5,918	1,658	431	641	152	502	171	1,603	875
49.5%	13.9%	3.6%	5.4%	1.3%	4.2%	1.4%	13.4%	7.3%
PCEDT								
10,226	1,256	305	273	435	633	96	845	1,375
66.2%	8.1%	2.0%	1.8%	2.8%	4.1%	0.6%	5.5%	8.9%
Project Syndicate (Named entities)								
8,978	1,781	76	420	165	641	48	1,165	317
66.1%	13.1%	0.6%	3.1%	1.2%	4.7%	0.4%	8.6%	2.3%
Total								
41,073	9,007	1,076	2,374	1,031	2,827	490	6,614	3,344
60.5%	13.3%	1.6%	3.5%	1.5%	4.2%	0.7%	9.7%	4.9%

numbers there is also percentage for easier comparison. 100% equals to all links made at least by one annotator.

There are the inter-annotator agreement results in Table 4.4. For every data source three types of agreement were measured:

- *Types distinguished* – We distinguish types of connections here. In this case in $A_1 \cap A_2$ there are only links that both the annotators labeled equally.
- *Types not distinguished* – We do not distinguish types of connections. In $A_1 \cap A_2$ there are all connections that were labeled by both the annotators. It does not matter which connection type they used.
- *Sure connections only* – We deal only with sure connections. All other connections are taken as null connections.

We can see that the highest agreement reached the data from Acquis Communautaire corpus (the European laws), because the translation here have to be very precise and close. Conversely, the inter-annotator agreement is lower for the texts from books and from the magazine Reader's Digest, whose sentences are translated very freely.

Table 4.4: Inter-annotator agreement of manual word alignment

Data source	Inter-annotator agreement		
	Types distinguished	Types not dist.	Sure only
Acquis Communautaire	86.7 %	92.1 %	94.0 %
Project Syndicate	80.8 %	87.8 %	92.0 %
Reader's Digest, Kačenka, Books	76.6 %	85.8 %	90.1 %
PCEDT	84.1 %	89.8 %	93.8 %
Proj. Syndicate (Named Entities)	86.5 %	91.5 %	93.6 %
Total	**83.3 %**	**89.6 %**	**92.9 %**

4.4 Transferring Alignment to T-Trees

The manual word-alignment has to be transfered up to the generated tectogrammatical trees so that we have the data for t-aligner evaluation.

Every t-node has an attribute which can point to one word on the surface from which it got its lexical meaning. Two t-nodes are aligned, if their corresponding words on surface are aligned. Types of connections are the same as for word-alignment. Consequently, links connecting words which do not have respective node on tectogrammatical layer do not appear in the tectogrammatical alignment. It concerns mainly articles, prepositions, and other functional words that are generally connected by possible links.

Added t-nodes that do not have their corresponding words on surface (e.g. `#PersPron` representing personal pronouns in Czech t-trees) are more problematic. Figure 4.2 illustrates the correction of links that connect English `#PersPron` t-node with Czech verb. This correction is automatic and uses simple heuristic rules.

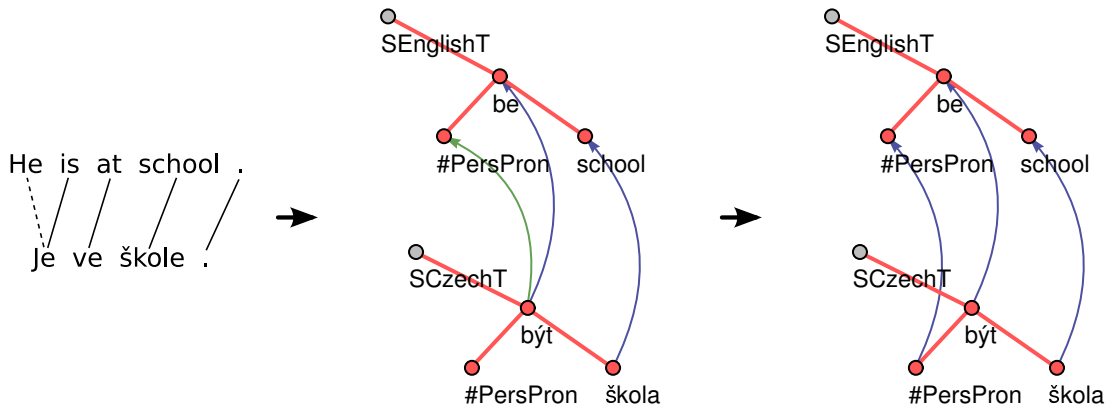


Figure 4.2: Correction of `#PersPron` connections

The same tables as for manual word-alignment were created for produced tectogrammatical alignment (tables 4.5, 4.6, and 4.7). We can see that there are fewer possible and phrasal links. Inter-annotator agreement increased. For example, if we do not distinguish types of connections, the total agreement raised from 89.6% up to 94.6%. This improvement supports our initial expectations about t-alignment.

Table 4.5: T-alignment transfered from manual word-alignment statistics

Data source	Sent.	Sure		Possible		Phrasal	
		A1	A2	A1	A2	A1	A2
Acquis Communautaire	501	6474	6694	439	285	3	53
Project Syndicate	484	5180	5648	619	442	19	13
Reader's Digest, Kačenka, Books	500	4016	4436	706	726	18	50
PCEDT	515	7173	7351	30	60	62	16
Project Syndicate (Named Entities)	500	7012	7103	170	170	3	30
Total	2500	29855	31232	1964	1683	105	162

Table 4.6: Occurrences of annotator agreement and disagreement for t-alignment

sure – sure	possible – possible	phrasal – phrasal	sure – possible	sure – phrasal	sure – no link	possible – phrasal	possible – no link	phrasal – no link
Acquis Communautaire								
6,297	122	0	250	31	293	1	229	24
86.9%	1.7%	0.0%	3.4%	0.4%	4.0%	0.0%	3.2%	0.3%
Project Syndicate								
5,073	164	0	317	9	356	0	416	23
79.8%	2.6%	0.0%	5.0%	0.1%	5.6%	0.0%	6.5%	0.4%
Reader's Digest, Kačenka, Books								
3,852	265	6	363	12	373	3	536	41
70.7%	4.9%	0.1%	6.7%	0.2%	6.8%	0.1%	9.8%	0.8%
PCEDT								
6,920	4	3	32	31	621	1	49	40
89.9%	0.1%	0.0%	0.4%	0.4%	8.1%	0.0%	0.6%	0.5%
Project Syndicate (Named entities)								
6,802	28	1	143	15	353	3	138	13
90.7%	0.4%	0.0%	1.9%	0.2%	4.7%	0.0%	1.8%	0.2%
Total								
28,944	583	10	1105	98	1,996	8	1,368	141
84.5%	1.7%	0.0%	3.2%	0.3%	5.8%	0.0%	4.0%	0.4%

Table 4.7: Inter-annotator agreement of t-alignment transfered from manual word-alignment

Data source	Inter-annotator agreement		
	Types distinguished	Types not dist.	Sure only
Acquis Communautaire	92.0 %	96.1 %	95.6 %
Project Syndicate	87.9 %	93.3 %	93.7 %
Reader's Digest, Kačenka, Books	82.9 %	90.5 %	91.2 %
PCEDT	94.3 %	95.1 %	95.3 %
Proj. Syndicate (Named Entities)	94.3 %	96.5 %	96.4 %
Total	**90.9 %**	**94.6 %**	**94.8 %**

Chapter 5

Implementation of Tectogrammatical Tree Aligner

In this chapter our new aligner of tectogrammatical trees will be described. It was developed in the TectoMT framework which was introduced in Chapter 3. The alignment process consists of two phases. In the first phase (Section 5.2) 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 (Section 5.4) other connections are added. Simple algorithm finds unaligned t-nodes and align them with already aligned t-nodes in the other language, if certain conditions are fulfilled.

The algorithm produces only one type of connections. Every t-node is aligned with no, one or more t-nodes in the opposite language. Phrasal alignment (N:N connections) is not implemented.

5.1 Preprocessing

Czech-English sentence pairs can be acquired from a parallel corpus. In the CzEng corpus [Bojar et al., 2008] there are tools for extracting 1:1 sentence pairs. Sentences are either tokenized or not. In many cases it is necessary to re-tokenize them according to the same tokenization rules. Slightly modified Penn Treebank style tokenization [Marcus et al., 1993] is used for English. Czech tokenizer is very simple – each non-alphanumeric and non-white character is an extra token and all alphanumeric sequences (words) are tokens.

After that follows the tectogrammatical analysis, which was described in Section 3.3. Czech sentences are morphologically analyzed and disambiguated by the morphological tagger shipped with PDT 2.0 [Hajič et al., 2006], syntactically analyzed by McDonald's MST parser [McDonald et al., 2005], and the analytical trees are converted into tectogrammatical trees by software components already available in TectoMT. English sentences are tagged by the TnT tagger [Brants, 2000], syntactically analyzed by the Collins parser [Collins, 1999], created phrase trees are converted into dependencies and finally into the tectogrammatical trees.

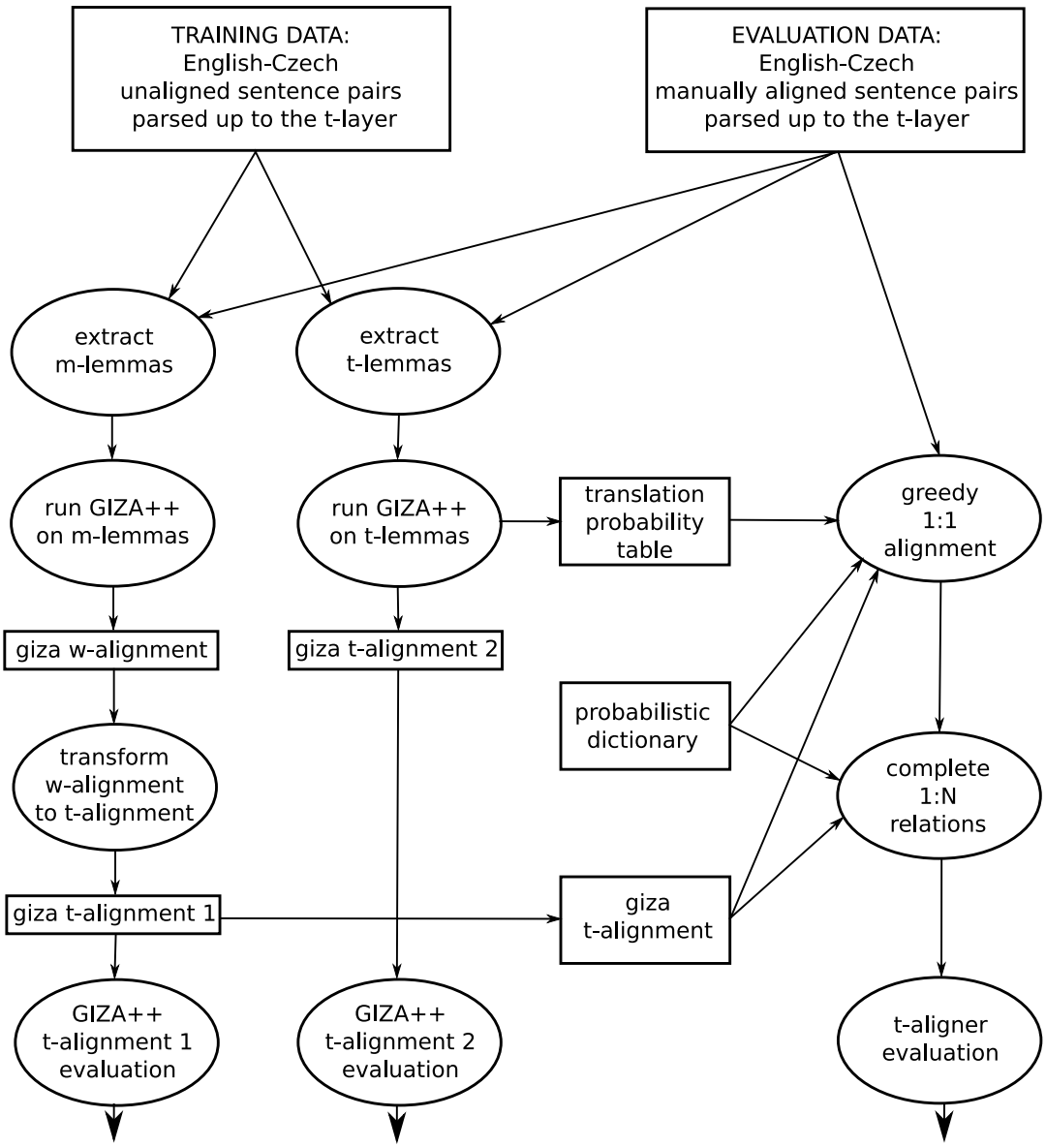


Figure 5.1: Data flow diagram of t-alignment and its evaluation

Then, the manual word-alignment is transfered to the generated tectogrammatical trees. This was described in Section 4.4.

Tectogrammatical trees are now ready to be aligned. But experiments showed that it is good to make one more thing before aligning process – align trees by GIZA++ tool first [Och and Ney, 2003]. If the t-aligner uses also the GIZA++ output, the results are slightly better. Principles of GIZA++ were described in Section 2.2. We have two possibilities how to align t-trees with GIZA++:

1. *direct t-alignment* – T-lemmas are extracted from the tectogrammatical trees and ordered according to their *deepord* attribute. These sequences are then processed by GIZA++. Note that there is no information about the tree structure or other attributes. However, the f-measure of this t-alignment reaches about 84%.

2. *t-alignment transferred from w-alignment* – Lemmatized sentences are aligned by GIZA++ on the surface. The resulting word-alignment is then transfered to the tectogrammatical trees in the same way as in Section 4.4. This t-alignment f-measure is higher – almost 86%.

Our t-aligner uses the second variant because of the higher f-measure. However, the first variant is used for generating t-lemma translation probability table. Experiments with GIZA++ will be described in Section 6.4.

There is the t-aligner data flow diagram in Figure 5.1. It includes also GIZA++ preprocessing and evaluation.

5.2 Greedy Algorithm for 1:1 Alignment

The first phase is based on a linear model and was inspired by the article [Menezes and Richardson, 2001]. Consider all potential alignment pairs between two trees. 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 tectogrammatical node, e_j is the $j-$th English tectogrammatical node, $\overrightarrow{w}$ is the vector of feature weights, and $\overrightarrow{f}$ is the vector of feature values. The features are listed in Section 5.3. The set of features was designed manually.

Pseudo-code of the algorithm is given in Figure 5.2. 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 recompute some pair scores after each step, because some features might be influenced by the already aligned pairs.

For the first time the weights were assigned to the features manually. Afterwards, we used an implementation of the discriminative reranker described in [Collins, 2002] and implemented by Václav Novák for optimizing the weights. The reranker is based on a modified perceptron algorithm.

Input: *TreePairs* – Czech and English tectogrammatical trees
Output: Aligned tectogrammatical trees

foreach (*CT, ET*) ∈ *TreePairs* **do**
  foreach *cnode* ∈ *CT* **do**
    used(*cnode*) = 0;
    foreach *enode* ∈ *ET* **do**
      used(*enode*) = 0;
      score(*cnode, enode*) = $\overrightarrow{w} \cdot \overrightarrow{f}$ (*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*) = $\overrightarrow{w} \cdot \overrightarrow{f}$ (*cnode, enode*);
    else
      break;

Figure 5.2: First phase of t-alignment in pseudo-code

5.3 Features

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

Several features use besides information about t-tree structure and attributes of t-nodes also other three sources:

a) Probabilistic dictionary – This dictionary was compiled from parallel corpora PCEDT [Cuřín et al., 2004]. Afterwards it was extended by word pairs acquired from parallel corpus CzEng [Bojar et al., 2008] aligned on word layer.

b) GIZA++ t-lemma alignment – Two features examine whether the examined pair of t-nodes were also aligned by GIZA++ or not. Intersection and grow-diag-final symmetrization method are used for this purposes.

c) GIZA++ translation probability table – Besides the alignment GIZA++ also produce several tables including the translation probability table which is used by one of the features.

Features can return a binary, integer, or real value. The list of features used follows:

- **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, otherwise equal to 0.
- **aligned by GIZA++, grow-diag-final** (*binary*) – Equal to 1 if the two nodes were aligned by GIZA++ with the grow-diag-final symmetrization, otherwise equal to 0.
- **translation probability from GIZA++** (*real*) – Returns the mean of t-lemma translation probabilities in both directions that were acquired from GIZA++ output translation tables.

$$p_{giza}(e_i, c_j) = \frac{p(t\_lemma(e_i) \mid t\_lemma(c_j)) + p(t\_lemma(c_j) \mid t\_lemma(e_i))}{2}$$

- **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, otherwise equal to 0.

- **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 tectogrammatical trees.

$$sim(e_i, c_j) = 1 - \left| \frac{i}{|e|} - \frac{j}{|c|} \right|$$

5.4 Algorithm for Completing 1:N Alignments

In the second 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 $parent(K)$ 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 $child(M)$ 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 probabilistic dictionary (no matter how high its translation probability is), the algorithm align this pair of t-nodes. There is a pseudo-code in Figure 5.3.

The described procedure was created experimentally. Combination of probabilistic dictionary with the GIZA++ t-alignment brought the highest improvement in f-measure.

Input: *AlignedTreePairs* – Partially aligned Czech and English tectogrammatical trees
Output: Aligned tectogrammatical trees

```
foreach (CT, ET) ∈ AlignedTreePairs do
    foreach cnode ∈ CT do
        foreach enode ∈ ET do
            if aligned(cnode, enode) then
                used(cnode) = 1;
                used(enode) = 1;
    foreach cnode ∈ CT do
        foreach enode ∈ ET do
            is_candidate = 0;
            if not used(cnode) then
                if aligned(parent(cnode), enode) then
                    is_candidate = 1;
                foreach c_child ∈ children(cnode) do
                    if aligned(c_child, enode) then
                        is_candidate = 1;
            if not used(enode) then
                if aligned(cnode, parent(enode)) then
                    is_candidate = 1;
                foreach e_child ∈ children(enode) do
                    if aligned(cnode, e_child) then
                        is_candidate = 1;
            if is_candidate and aligned_by_giza_gdf(cnode, enode) and
            is_in_dictionary(tlemma(cnode), tlemma(enode)) then
                Align(cnode, enode);
```

Figure 5.3: Second phase of t-alignment in pseudo-code

Chapter 6

Experiments and Results

This chapter concerns the evaluation of implemented tectogrammatical aligner. It contains many tables that compare alignment qualities depending on the type of data at the input (texts from laws, newspaper articles, stories). In Section 6.4 there are tables concerning experiments with GIZA++ and with various methods of symmetrization.

6.1 Evaluation Process

All the data that were manually aligned on word layer were used for evaluation of the t-aligner and for training weights of the features. They were automatically analyzed up to the t-layer and word-alignment was transferred into the alignment of t-nodes as described in Section 4.4. Each sentence is aligned by two annotators. The golden alignment was thus created from the two parallel annotations 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. This merging of two alignments was also used in [Bojar and Prokopová, 2006].

There are three possibilities how to deal with the golden alignment. There are two types of connections – *sure* and *possible* ones, while our structural t-aligner makes only one type of connection. Three following evaluation variants present themselves:

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

3. *possible do not mind* – If there is a possible connection in the golden alignment, it does not matter whether t-aligner makes here a connection or not. Possible connections are not included in evaluation calculation.

Table 6.1: Comparison of results for the two evaluation variants

evaluation method	precision	recall	f-measure
both types	94.31 %	81.62 %	87.51 %
sure only	92.50 %	89.63 %	91.04 %
possible do not mind	96.01 %	89.67 %	92.73 %

We decided to use the *sure only* variant for all evaluations. The other variants *both types* and *possible do not mind* were used only once for comparison of this three methods. The differences are depicted in Table 6.1. There are the results for all evaluation data (2500 sentences).

We can see that the results for *both types* evaluation variant are worse than for *sure only* variant. It is caused mainly by the fact that the golden alignment was created by merging two alignments and there are too many possible connections. The major part is the set of connections that were made by one annotator only. This implies the low recall.

The third evaluation variant *possible do not mind* solves this problem of possible connections. It does not matter whether t-aligner makes here a connection or not, if the connection is *possible*. The results for this evaluation variant are better than for *both types* variant. The disadvantage is that this variant does not include into calculation all the connections. The f-measure would raise with the increasing rate of possible connections and this is not desirable.

6.2 Cross-validation Results for Various Types of Data

We used 10-fold cross-validation method for the t-aligner evaluation. The process is repeated ten times, each tenth of the data is used exactly once for validation. The remaining nine tenths of the data are used for training the feature weights and for the optimal threshold setting. Precision, recall and f-measure are computed in each iteration. Precision indicates the percentage of how many pairs aligned by this algorithm were aligned also by annotator; recall indicates how many pairs aligned by the annotator were aligned by the algorithm. F-measure is their harmonic mean. The values of precision recall and f-measure from all iterations are then averaged.

$$fmeasure = \frac{2 \cdot precision \cdot recall}{(precision + recall)}$$

We split the evaluation data into 5 groups as it was done for word-alignment evaluation (Section 4.3): *Acquis Communautaire* in the first group, *Project Syndicate* in the second group, *Reader's Digest*, *Kačenka*, and *E-books* in third, sentences from *PCEDT* in fourth, and last group contains sentences with named entities from *Project Syndicate*. You can see the results in tables 6.2, 6.3, 6.4, 6.5, and 6.6. Final f-measures are bold.

Table 6.2: 10-fold cross-validation results for data from Acquis Communautaire

n	1	2	3	4	5	6	7	8	9	10	mean
P	91.13	94.21	92.68	94.12	89.91	94.01	93.82	94.54	89.65	94.13	92.82
R	83.90	93.41	91.67	93.23	93.06	92.21	93.56	92.88	88.39	91.76	91.41
F	87.37	93.81	92.17	93.67	91.46	93.10	93.69	93.70	89.02	92.93	**92.09**

Table 6.3: 10-fold cross-validation results for data from Project Syndicate

n	1	2	3	4	5	6	7	8	9	10	mean
P	90.33	92.13	95.84	98.05	88.22	86.55	86.48	89.05	90.74	90.05	90.74
R	90.88	90.19	95.33	95.28	92.09	91.07	88.27	89.81	92.21	93.47	91.86
F	90.61	91.15	95.58	96.64	90.12	88.75	87.36	89.43	91.47	91.73	**91.28**

Table 6.4: 10-fold cross-validation results for data from Reader's Digest, Books and Kačenka

n	1	2	3	4	5	6	7	8	9	10	mean
P	84.39	86.06	84.95	87.16	89.86	81.72	88.04	85.86	88.25	86.55	86.28
R	82.59	81.84	79.97	85.51	84.68	76.99	83.00	84.57	87.83	79.83	82.68
F	83.48	83.89	82.38	86.33	87.19	79.28	85.44	85.21	88.04	83.06	**84.43**

Table 6.5: 10-fold cross-validation results for data from PCEDT

n	1	2	3	4	5	6	7	8	9	10	mean
P	95.41	95.17	95.30	97.30	90.33	94.59	97.03	93.87	93.94	95.50	94.85
R	88.71	90.59	91.48	92.07	81.02	88.22	90.28	90.69	83.32	90.95	88.73
F	91.94	92.82	93.35	94.62	85.43	91.29	93.53	92.25	88.31	93.17	**91.67**

Table 6.6: 10-fold cross-validation results for data from Project Syndicate (Named Entities)

n	1	2	3	4	5	6	7	8	9	10	mean
P	93.00	95.21	95.34	96.76	96.41	93.71	93.85	94.77	95.87	93.47	94.84
R	91.79	91.48	93.62	91.11	92.73	92.48	92.57	94.62	93.47	93.65	92.75
F	92.39	93.31	94.47	93.85	94.54	93.09	93.21	94.69	94.65	93.56	**93.78**

Table 6.7: 10-fold cross-validation results for all evaluation data together

n	1	2	3	4	5	6	7	8	9	10	mean
P	91.75	92.99	92.70	95.18	91.32	90.64	92.10	92.96	92.26	93.06	92.50
R	88.25	89.43	90.15	91.43	87.88	88.06	89.45	91.52	88.82	91.34	89.63
F	89.97	91.18	91.41	93.27	89.57	89.33	90.76	92.23	90.51	92.19	**91.04**

In Table 6.7 there are the 10-fold cross-validation results for all data that were manually aligned (2500 sentences). Different types of data were distributed uniformly into 10 groups.

The differences of f-measures for various types of data correspond to our expectations. The lowest f-measure (84.43%) was computed for literary texts – the data set *Reader's Digest, Books and Kačenka*. This texts were translated very freely; sometimes even sentences do not match, whence it follows that to align them is problematic and f-measure will be low. On the other side, texts from the data set *Acquis Communautaire* reached 92.09% f-measure. Law texts are translated very precisely and literally, a lot of words have their own equivalents and therefore the alignment is easier. The highest f-measure was reached by the set *Project Syndicate (Named Entities)*. Named entities (e.g. names of persons, countries, corporations etc.) have very simple alignment, mostly 1:1 non-crossing connections. Their enhanced occurrence increased the f-measure from 91.28% (common sentences from *Project Syndicate*) to 93.78% (sentences with named entities).

6.3 Weights of Features

In Table 6.8 there are the feature weights that were estimated by perceptron in one of the training iterations. For all data the acquired weights were similar. All feature values are either binary (0 or 1) or probabilistic (between 0 and 1). The only exception is the feature *aligned child*, whose value can be $\{0, 1, 2, \dots\}$. Thus we can say the weights are normalized and we can order them according to their importance.

Besides the weight vector, also the threshold value is needed in the algorithm. Its value was found by hill-climbing method after the feature weights were estimated. Its optimal value for weights given in Table 6.8 is 3.40 for *sure only* evaluation variant. For the variants *both types* and *possible do not mind* it is 3.00 and 3.15 respectively.

There is an example of Czech-English aligned trees in Figure 6.1. In this case, t-aligner made no errors, but there were some errors in the built trees. 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 6.8: Feature weights obtained by the perceptron

feature	values	weight
similarity in linear position	$\langle$ 0, 1 $\rangle$	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	$\langle$ 0, 1 $\rangle$	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.	$\langle$ 0, 1 $\rangle$	0.17
same shortened formeme	0 or 1	0.11

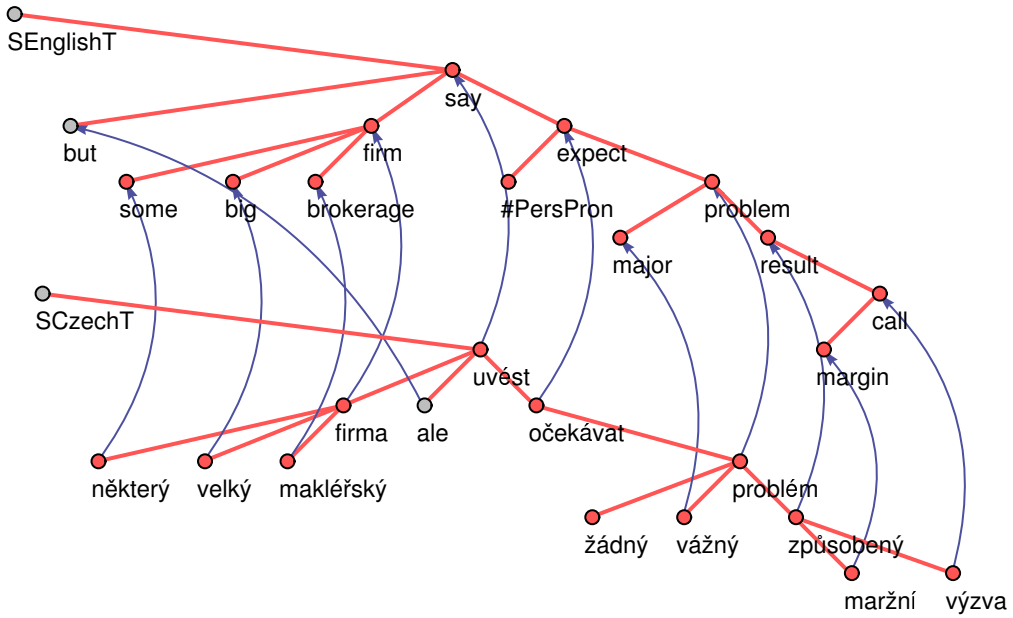


Figure 6.1: Tectogrammatical tree alignment of the sentence *"But some big brokerage firms said they don't expect major problems as a result of margin calls."*, the Czech translation is *"Některé velké makléřské firmy ale uvedly, že neočekávají žádné vážné problémy způsobené maržními výzvami."*

6.4 Experiments with GIZA++ Alignment Tool

All the evaluation data were also aligned on tectogrammatical layer by GIZA++. We evaluated both the t-alignment variants described in Section 5.1.

In the first variant ("direct t-alignment") the sequences of *t-lemmas* are extracted and ordered according to their *deepord* attribute. Each t-node is represented by one *t-lemma*. This sequences are then aligned by GIZA++. Since GIZA++ tool aligns sentences in one direction only, it was run twice in both Czech-to-English and English-to-Czech directions and then symmetrized by one of the symmetrization method described in Subsection 2.2.2. You can see the results in Table 6.9. The best f-measure was accomplished by the intersection symmetrization. All data and *sure only* evaluation variant were used. The results depending on the type of data are in Table 6.10.

In the second variant ("t-alignment transfered from w-alignment") the lemmatized sentences are first aligned by GIZA++ and the word alignment is afterwards transfered to the tectogrammatical alignment. The evaluation results of this variant are in Table 6.11 and Table 6.12. This variant outperforms the first one. It is caused probably by the fact that GIZA++ is optimized for word-level alignment.

Table 6.9: GIZA++ "direct t-alignment" results depending on the symmetrization method. All the evaluation data (2500 sentences) were used.

Symmetrization type	Precision	Recall	F-measure
Source to Target	73.05 %	84.79 %	78.48 %
Target to Source	71.33 %	75.26 %	73.24 %
Union	60.84 %	91.56 %	73.11 %
Intersection	93.10 %	75.93 %	83.64 %
Grow	75.67 %	81.66 %	78.55 %
Grow-diag	71.20 %	87.10 %	78.35 %
Grow-diag-final	63.82 %	90.78 %	74.95 %
Grow-diag-final-and	70.29 %	88.46 %	78.33 %

Table 6.10: GIZA++ "direct t-alignment" results depending on the data source. Intersection symmetrization method was used.

Data source	Precision	Recall	F-measure
Acquis Communautaire	93.46 %	83.74 %	88.33 %
Project Syndicate	93.79 %	80.88 %	86.86 %
Reader's Digest, Kačenka, Books	85.32 %	53.91 %	66.07 %
PCEDT	93.25 %	74.16 %	82.62 %
Project Syndicate (Named entities)	95.68 %	80.20 %	87.27 %
Total	**93.10 %**	**75.93 %**	**83.64 %**

Table 6.11: GIZA++ "t-alignment transferred from w-alignment" results depending on the symmetrization method. All the evaluation data (2500 sentences) were used.

Symmetrization type	Precision	Recall	F-measure
Source to Target	78.51 %	86.06 %	82.11 %
Target to Source	75.87 %	79.22 %	77.51 %
Union	66.60 %	92.91 %	77.58 %
Intersection	95.45 %	77.75 %	85.70 %
Grow	83.89 %	82.80 %	83.34 %
Grow-diag	79.93 %	87.79 %	83.68 %
Grow-diag-final	71.08 %	91.64 %	80.06 %
Grow-diag-final-and	79.11 %	89.34 %	83.91 %

Table 6.12: GIZA++ "t-alignment transferred from w-alignment" results depending on the data source. Intersection symmetrization method was used.

Data source	Precision	Recall	F-measure
Acquis Communautaire	95.28 %	83.44 %	88.96 %
Project Syndicate	95.30 %	81.74 %	88.00 %
Reader's Digest, Kačenka, Books	90.08 %	59.18 %	71.41 %
PCEDT	96.83 %	76.58 %	85.52 %
Project Syndicate (Named entities)	97.00 %	82.07 %	88.91 %
Total	**95.45 %**	**77.75 %**	**85.70 %**

Chapter 7

Conclusions

T-aligner – the tool for aligning tectogrammatical trees was implemented in TectoMT Framework. The presented algorithm is 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.

For evaluation of the t-aligner manual annotations were realized. 2500 sentences were aligned manually on word layer, each part of data was aligned by two annotators. Annotators used three types of connections. Before and throughout the annotations the rules for most frequent phenomena were designed. Inter-annotator agreement was computed for all types of data.

Manual word alignment was transfered up to the tectogrammatical trees in order to be used as golden data for the t-aligner. Two different alignments from two annotators were merged together. Inter-annotator agreement was computed also for the tectogrammatical layer. The agreement here is higher than the agreement on the word layer.

The t-aligner was evaluated on five types on data. The resulting f-measure for all the data reached 91.0%. This result is still well below the upper limit – the inter-annotator agreement on t-layer alignment reaches 94.8% –, but outperforms the t-alignment derived from the alignment produced by GIZA++, the f-measure of which is 85.7%

Table 7.1 summarizes the baseline (alignment by GIZA++), the upper limit given by inter-annotator agreement, and the performance of the implemented t-aligner.

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

Table 7.1: Alignment evaluation summary (f-measure)

Data type	IAA W-layer	IAA T-layer	GIZA++ T-layer	t-aligner T-layer
Acquis Communautaire	94.0 %	95.6 %	89.0 %	92.1 %
Project Syndicate	92.0 %	93.7 %	88.0 %	91.3 %
Reader's Digest, Kačenka, Books	90.1 %	91.2 %	71.4 %	84.4 %
PCEDT	93.8 %	95.3 %	85.5 %	91.7 %
Project Syndicate (Named entities)	93.6 %	96.4 %	88.9 %	93.8 %
Total	**92.9 %**	**94.8 %**	**85.7 %**	**91.0 %**

Bibliography

[Al-Onaizan et al., 1999] Al-Onaizan, Y., Cuřín, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F.-J., Purdy, D., Smith, N. A., and Yarowsky, D. (1999). Statistical Machine Translation. Technical report, JHU workshop.

[Böhmová et al., 2005] Böhmová, A., Cinková, S., and Hajičová, E. (2005). A Manual for Tectogrammatical Layer Annotation of the Prague Dependency Treebank (English translation). Technical report, ÚFAL MFF UK, Prague, Czech Republic.

[Bojar et al., 2007] Bojar, O., Cinková, S., and Ptáček, J. (2007). Towards English-to-Czech MT via Tectogrammatical Layer. In *Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories (TLT 2007)*, Bergen, Norway.

[Bojar et al., 2008] Bojar, O., Janíček, M., Žabokrtský, Z., Češka, P., and Beňa, P. (2008). CzEng 0.7: Parallel Corpus with Community-Supplied Translations. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. ELRA.

[Bojar and Prokopová, 2006] Bojar, O. and Prokopová, M. (2006). Czech-English Word Alignment. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1236–1239. ELRA.

[Bojar and Žabokrtský, 2006] Bojar, O. and Žabokrtský, Z. (2006). CzEng: Czech-English Parallel Corpus, Release version 0.5. *Prague Bulletin of Mathematical Linguistics*, 86:59–62.

[Brants, 2000] Brants, T. (2000). TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of the 6th Applied NLP Conference, ANLP-2000*, pages 224–231, Seattle.

[Brown et al., 1993] Brown, P. F., Pietra, S. D., Pietra, V. J. D., and Mercer, R. L. (1993). The Mathematic of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

[Brown et al., 1992] Brown, P. F., Pietra, V. J. D., de Souza, P. V., Lai, J., and Mercer, R. L. (1992). Class-based n-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479.

[Cinková et al., 2006] Cinková, S., Hajič, J., Mikulová, M., Mladová, L., Nedolužko, A., Pajas, P., Panevová, J., Semecký, J., Šindlerová, J., Toman, J., Urešová, Z., and Žabokrtský, Z. (2006). Annotation of English on the tectogrammatical level. Technical report, ÚFAL/CKL Technical Report TR-2006-35, MFF UK, Praha.

[Collins, 1999] Collins, M. (1999). *Head-driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, Philadelphia.

[Collins, 2002] Collins, M. (2002). Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, volume 10, pages 1–8.

[Cuřín et al., 2004] Cuřín, J., Čmejrek, M., Havelka, J., Hajič, J., Kuboň, V., and Žabokrtský, Z. (2004). Prague Czech-English Dependency Treebank, Version 1.0. Center for Computational Linguistics, Institute of Formal and Applied Linguistics, Linguistics Data Consortium (LDC) catalog number LDC2004T25, ISBN 1-58563-321-6, Prague.

[Hajič et al., 2006] Hajič, J., Hajičová, E., Panevová, J., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., and Mikulová, M. (2006). Prague Dependency Treebank 2.0. CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia.

[Hajičová et al., 1999] Hajičová, E., Kirschner, Z., and Sgall, P. (1999). A Manual for Analytic Layer Annotation of the Prague Dependency Treebank (English translation). Technical report, ÚFAL MFF UK, Prague, Czech Republic.

[Hana et al., 2005] Hana, J., Zeman, D., Hajič, J., Hanová, H., Hladká, B., and Jeřábek, E. (2005). Manual for Morphological Annotation, Revision for the Prague Dependency Treebank 2.0. Technical Report TR-2005-27, ÚFAL MFF UK, Prague, Czech Rep.

[Haruno and Yamazaki, 1996] Haruno, M. and Yamazaki, T. (1996). High-performance Bilingual Text Alignment Using Statistical and Dictionary Information. In *Proceedings of the 34th conference of the Association for Computational Linguistics*, pages 131–138, Santa Cruz, California.

[Koehn et al., 2007] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Demo and Poster Sessoins of the 45th Annual Meegting of the Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic.

[Marcus et al., 1993] Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

[McDonald et al., 2005] McDonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005). 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)*, pages 523–530, Vancouver, BC, Canada.

[Menezes and Richardson, 2001] Menezes, A. and Richardson, S. D. (2001). 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, pages 1–8.

[Och and Ney, 2000] Och, F. J. and Ney, H. (2000). Improved Statistical Alignment Models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistic*, pages 440–447, Honkong, China.

[Och and Ney, 2003] Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

[Ralf et al., 2006] Ralf, S., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., and Varga, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*.

[Rambousek et al., 1997] Rambousek, J., Chamonikolasová, J., Daniel Mikšík, D. S., and Kalivoda, M. (1997). KAČENKA (Korpus anglicko-český – elektronický nástroj Katedry anglistiky). http://www.phil.muni.cz/angl/kacenka.

[Sgall, 1967] Sgall, P. (1967). *Generativní popis jazyka a česká deklinace*. Academia, Prague, Czech Republic.

[Tinsley et al., 2007] Tinsley, J., Zhechev, V., Hearne, M., and Way, A. (2007). Robust Language Pair-Independent Sub-Tree Alignment. In *Proceedings of Machine Translation Summit XI*, pages 467–474, Copenhagen, Denmark.

[Watanabe et al., 2003] Watanabe, H., Kurohashi, S., and Aramaki, E. (2003). *Finding Translation Patterns from Paired Source and Target Dependency Structures*, pages 397–420. Kluwer Academic.

[Wu, 1997] Wu, D. (1997). Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377–403.

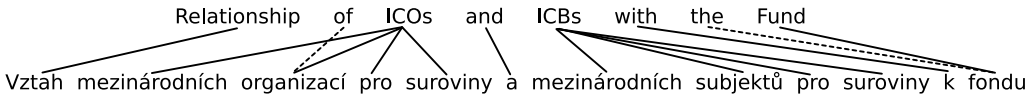
[Žabokrtský et al., 2008] Žabokrtský, Z., Ptáček, J., and Pajas, P. (2008). TectoMT: Highly Modular MT System with Tectogrammatics Used as Transfer Layer. In *Proceedings of WMT'08*.

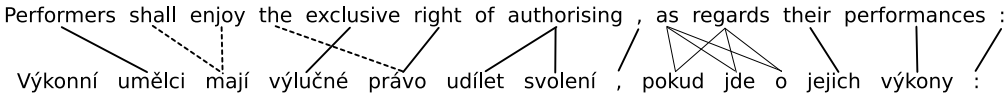
Appendix A

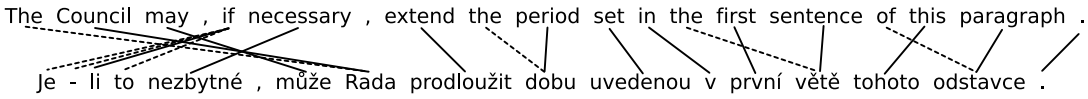
Examples of Word Alignment

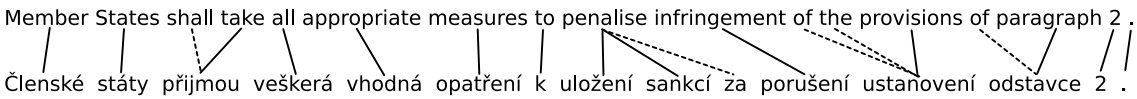
In this Appendix you can find examples of sentences that were manually aligned by annotators. They are divided into five groups according to their type. Three types of connections are distinguished as follows: bold solid lines are for *sure* connections, bold dashed lines for *possible* connections, and thin solid lines represent *phrasal* connections.

A.1 Sentences from Acquis Communautaire

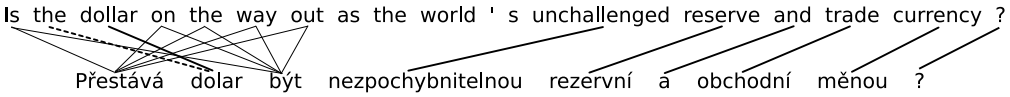


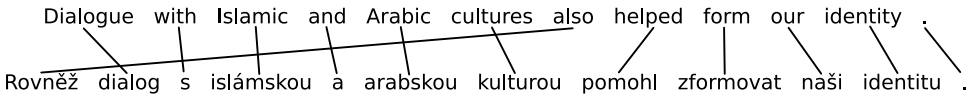


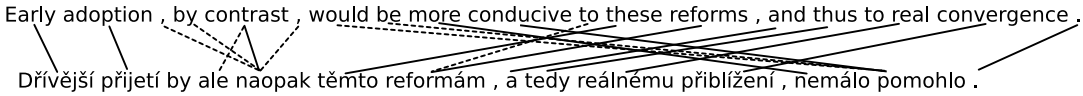


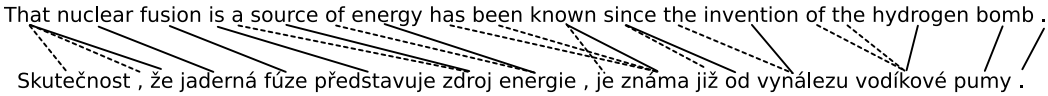


A.2 Sentences from Project Syndicate

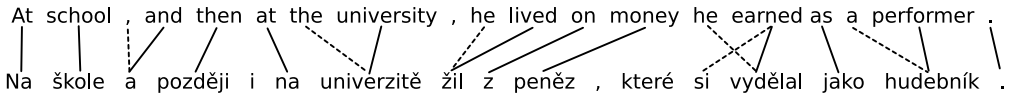


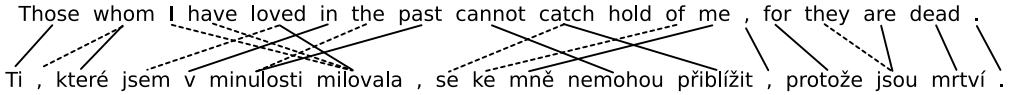


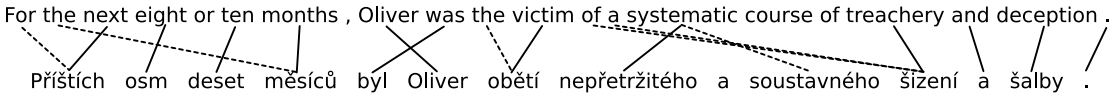


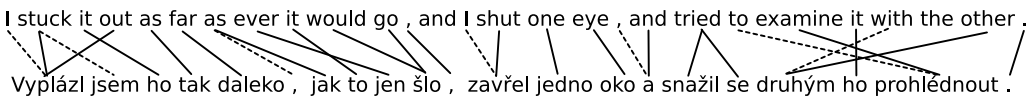


A.3 Sentences from Reader's Digest, Kačenka, E-Books

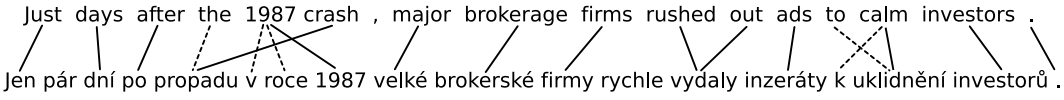


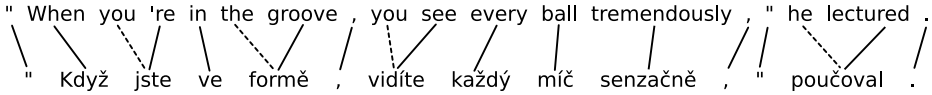


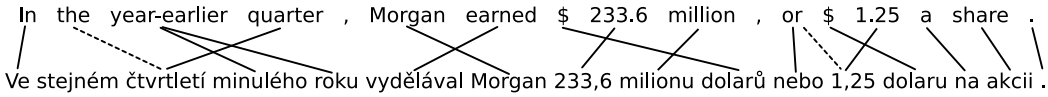


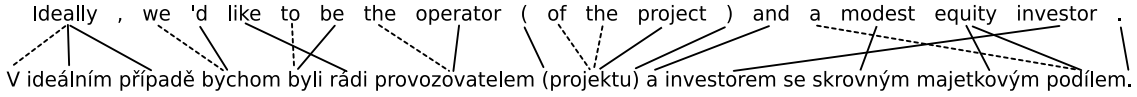


A.4 Sentences from PCEDT

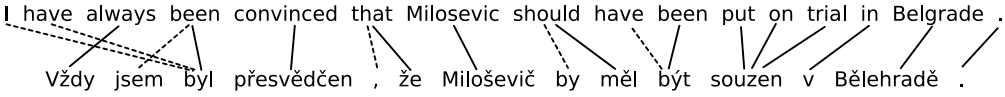


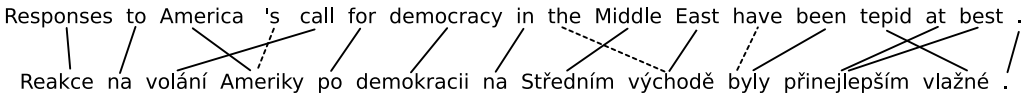


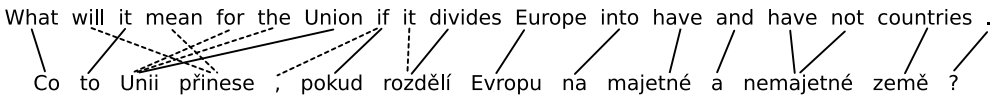


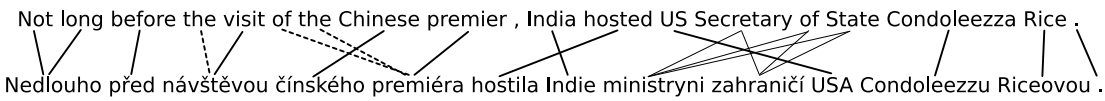


A.5 Sentences from Project Syndicate (Named Entities)







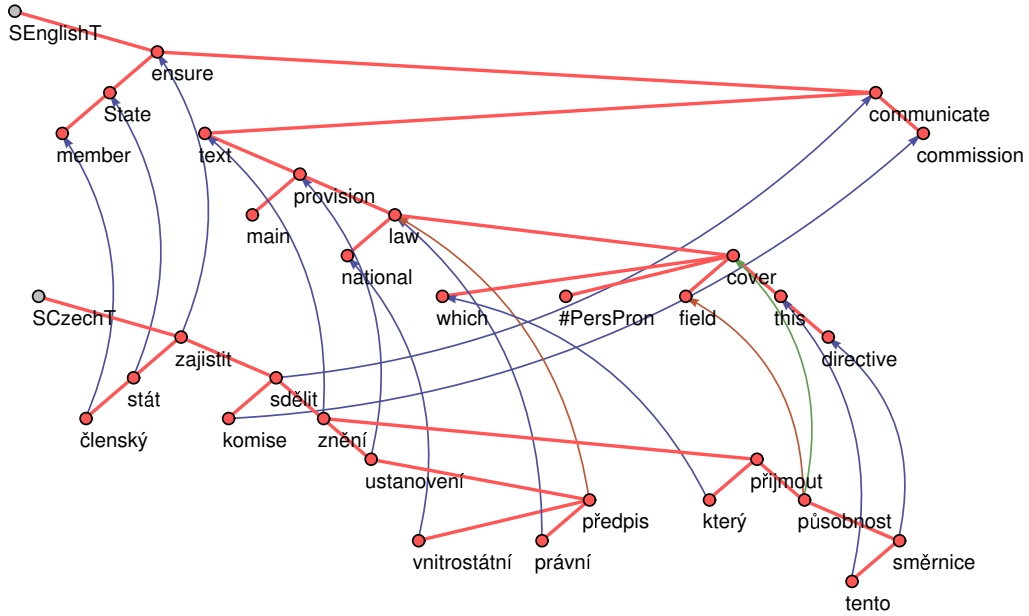


Appendix B

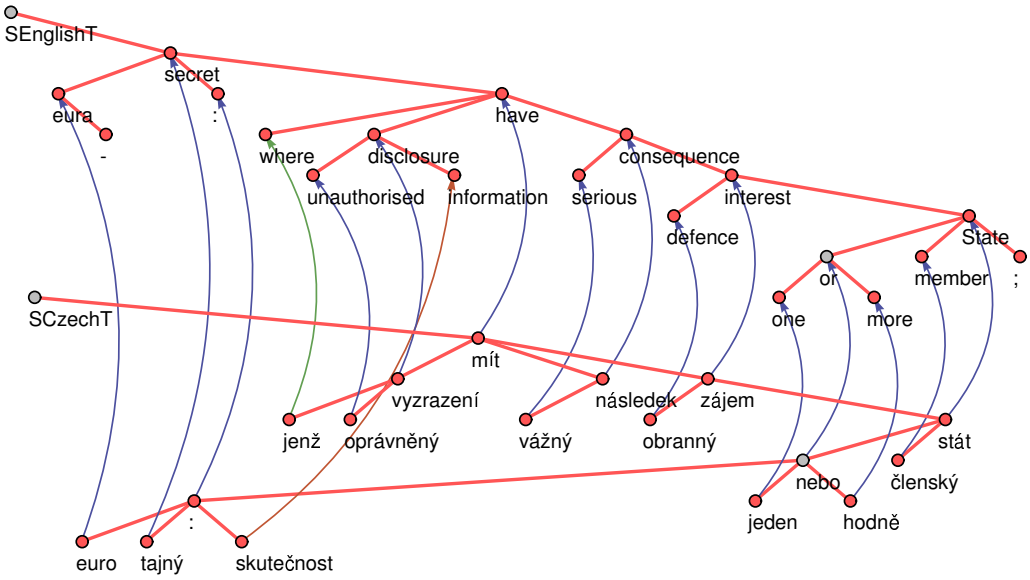
Examples of Aligned T-Trees

Here you can find examples of aligned Czech and English tectogrammatical trees made by our t-aligner. The examples are divided into five groups according to the type of source text. T-trees are simplified, only *t-lemmas* of t-nodes are depicted. There are three types of arrows. All represents *sure* connections only. The blue ones are connections made both by t-aligner and by annotator. Green connections made t-aligner but not annotator. Red ones were made by annotator only.

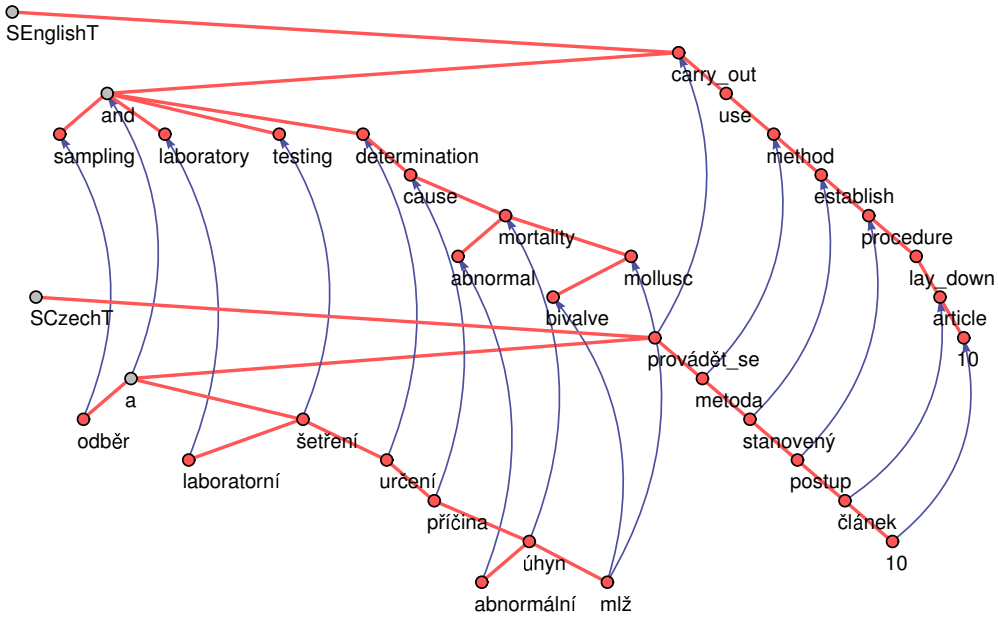
B.1 Sentences from Acquis Communautaire



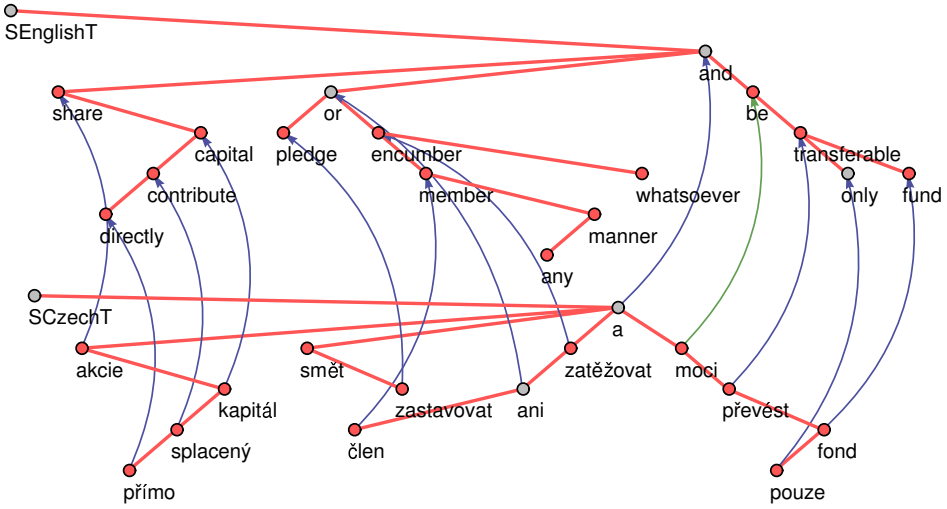
Member States shall ensure that the texts of the main provisions of national law which they adopt in the field covered by this Directive are communicated to the Commission.
Členské státy zajistí, aby bylo Komisi sděleno znění ustanovení vnitrostátních právních předpisů, které přijmou v oblasti působnosti této směrnice.



Eura – Secret: where unauthorised disclosure of the information would have serious consequences for the defence interests of one or more Member States;
Eura – Tajné: skutečnosti, jejichž neoprávněné vyzrazení by mohlo mít vážné následky pro obranné zájmy jednoho nebo více členských států;

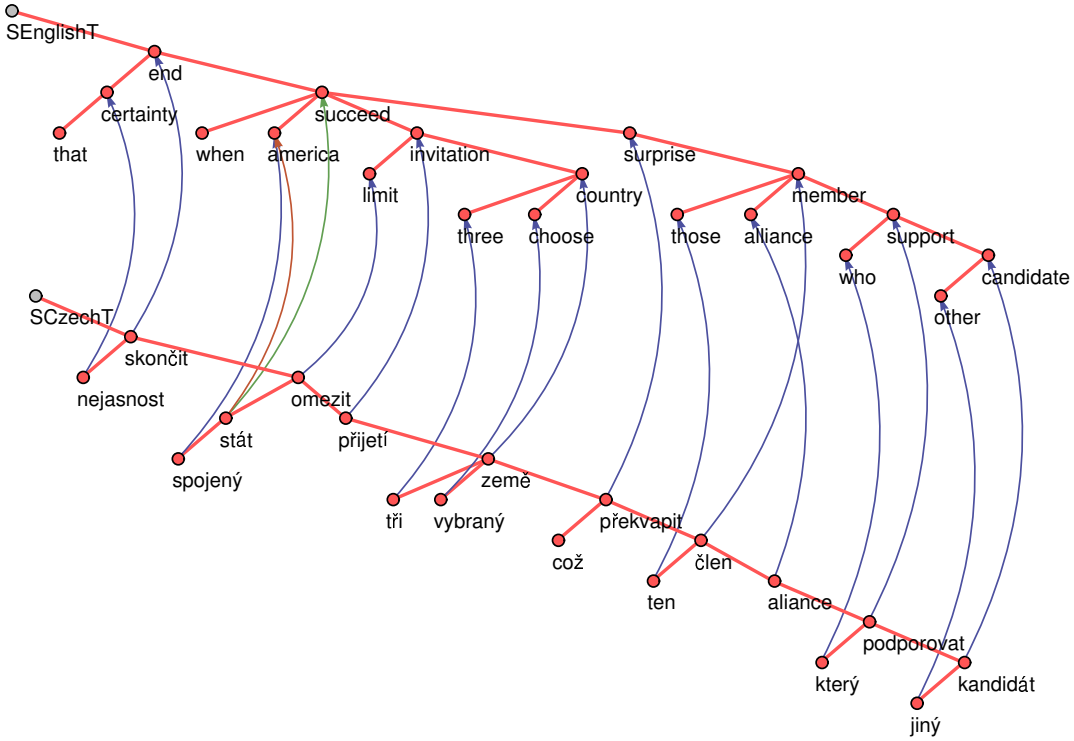


Sampling and laboratory testing for the determination of the cause of abnormal mortality of bivalve molluscs shall be carried out using the methods established in accordance with the procedure laid down in Article 10.
Odběry a laboratorní šetření k určení příčiny abnormálního úhynu mlžů se provádějí pomocí metod stanovených postupem podle článku 10.

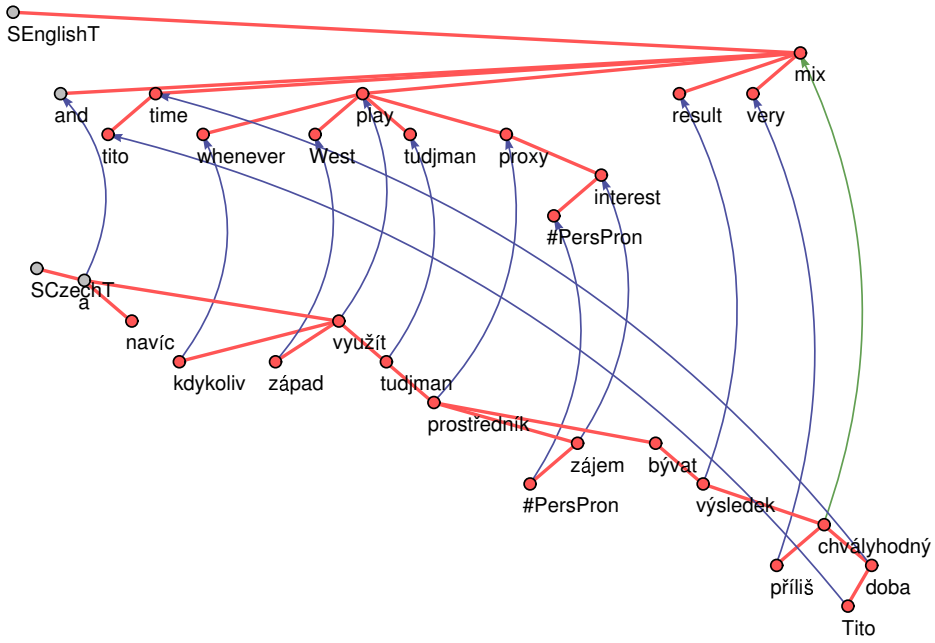


Shares of directly contributed capital shall not be pledged or encumbered by Members in any manner whatsoever and shall be transferable only to the Fund.
Akcie přímo splaceného kapitálu nesmějí být členy zastavovány ani zatěžovány a mohou být převedeny pouze na fond.

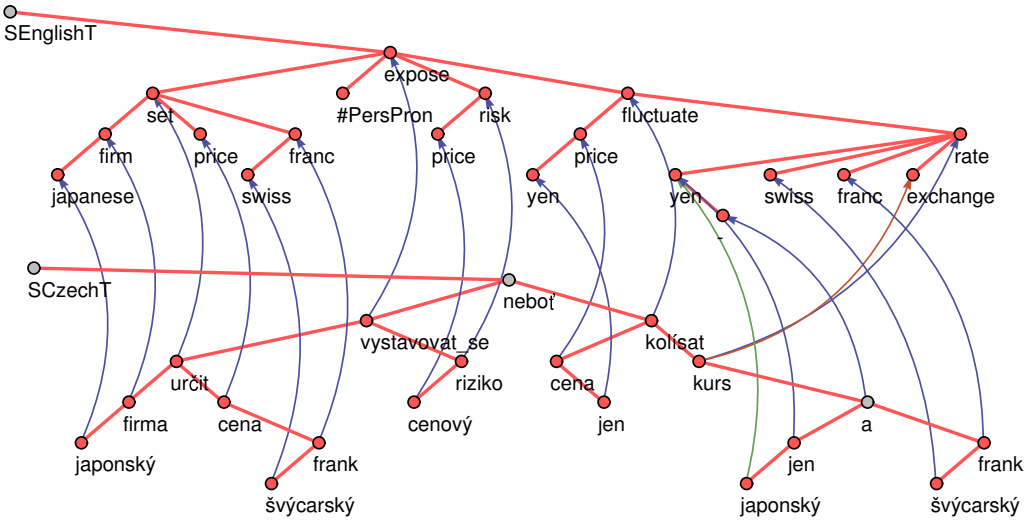
B.2 Sentences from Project Syndicate



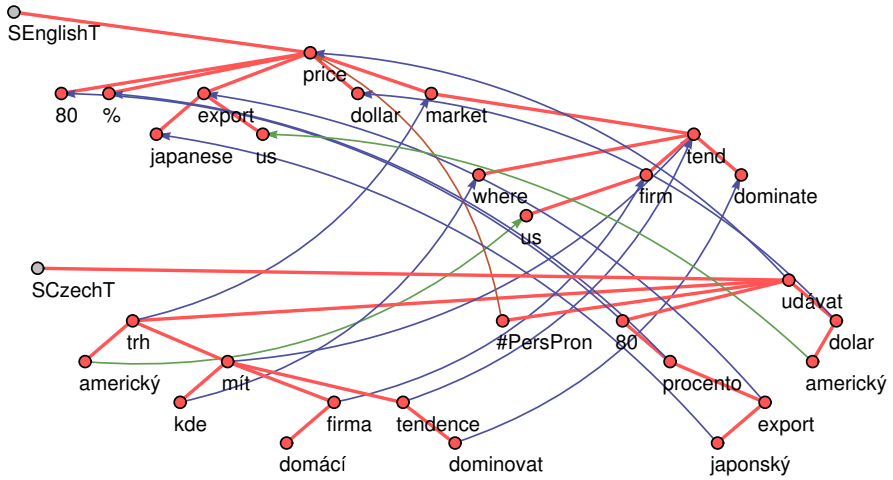
That uncertainty ended when America succeeded in limiting invitations to three chosen countries, surprising those Alliance members who supported other candidates.
Nejasnosti skončily, když Spojené státy omezily přijetí na tři vybrané země, čímž překvapily ty členy aliance, kteří podporovali jiné kandidáty.



And, unlike in Tito's time, whenever the West played Tudjman as a proxy for its interests the results were very mixed.
A navíc, kdykoli Západ využil Tudjmana jako prostředníka pro své zájmy, nebýval výsledek příliš chvályhodný – narozdíl od Titovy doby.

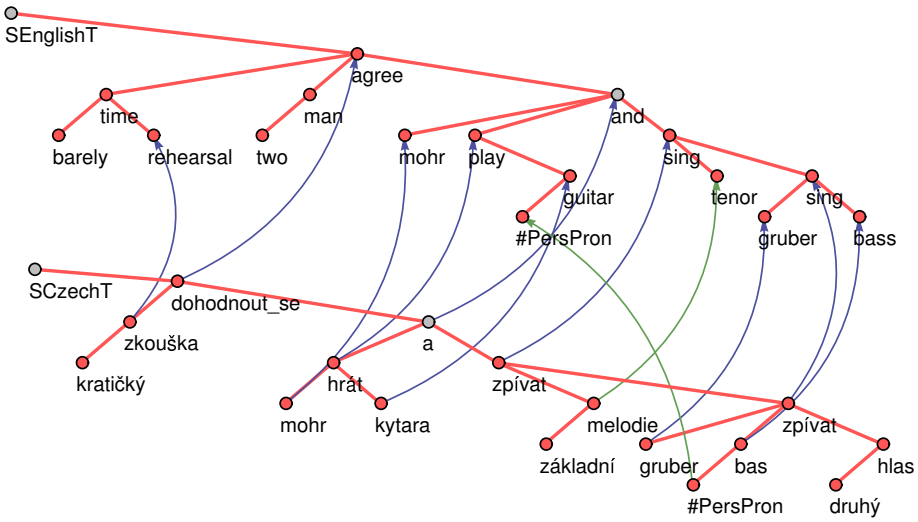


If the Japanese firm sets the price in Swiss francs, it is exposed to price risk as the yen price will fluctuate with the yen – Swiss franc exchange rate.
Pokud japonská firma určí cenu ve švýcarských francích, vystavuje se cenovému riziku, neboť cena v jenech bude kolísat podle kurzu mezi japonským jenem a švýcarským frankem.

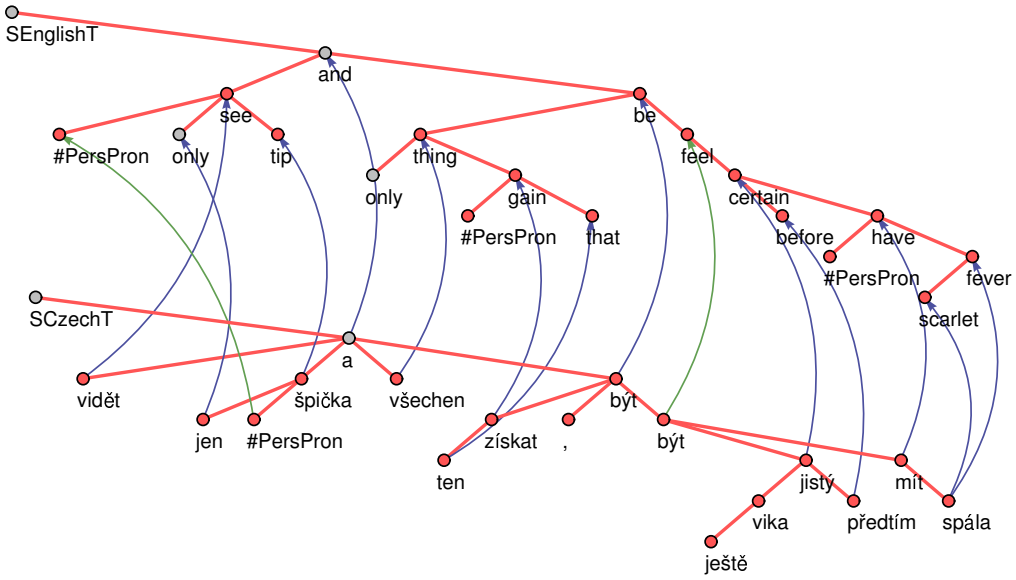


Over 80 % of Japanese exports to the US are priced in dollars, in markets where US firms tend to dominate.
Na amerických trzích, kde mají domácí firmy tendenci dominovat, se 80 procent japonského exportu udává v amerických dolarech.

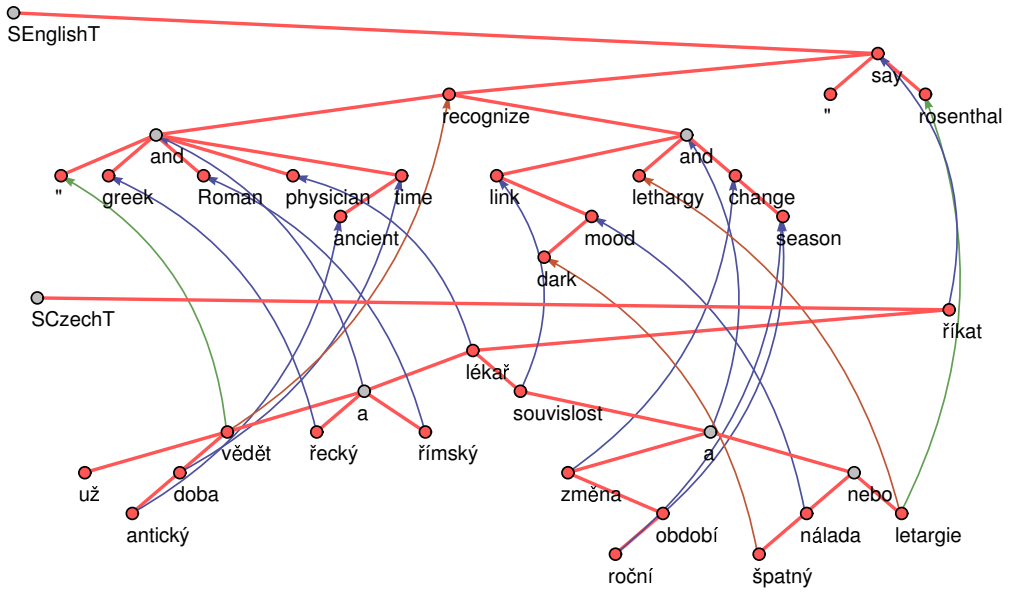
B.3 Sentences from Reader's Digest, Kačenka, E-Books



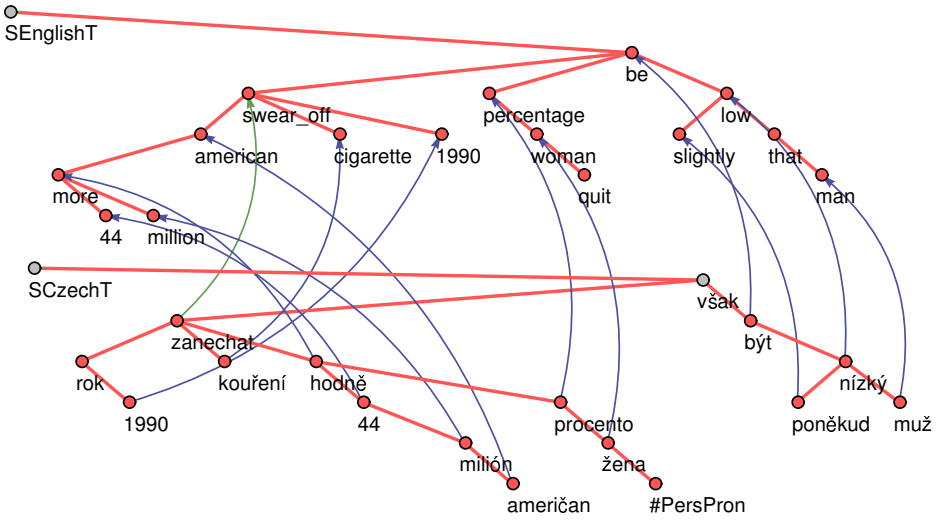
With barely time for a rehearsal, the two men agreed that Mohr would play his guitar and sing tenor while Gruber sang bass.
Při kratičké zkoužce se dohodli, že Mohr bude hrát na kytaru a zpívat základní melodii, zatímco Gruber bude svým basem zpívat druhý hlas.



I could only see the tip, and the only thing that I could gain from that was to feel more certain than before that I had scarlet fever.
Viděl jsem jen jeho špičku a všechno, co jsem z toho získal, bylo, že jsem si byl ještě více jist než předtím, že mám spálu.

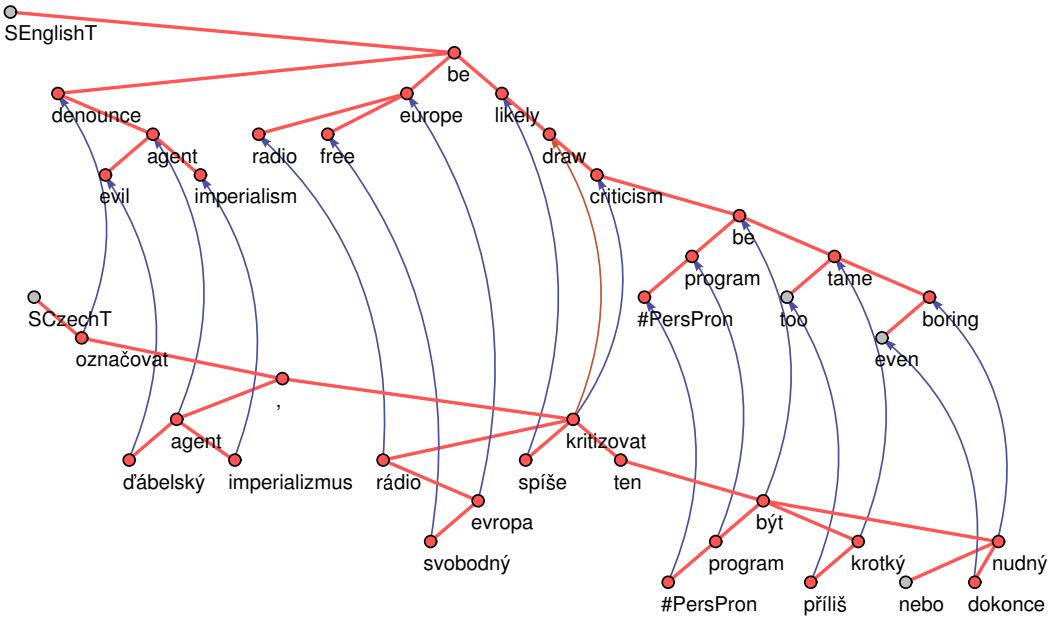


"Greek and Roman physicians in ancient times recognized a link between dark moods, lethargy and the change of seasons," says Rosenthal.
"Už v antických dobách věděli řečtí a římští lékaři o souvislosti mezi změnami ročních období a špatnou náladou nebo letargií," říká.

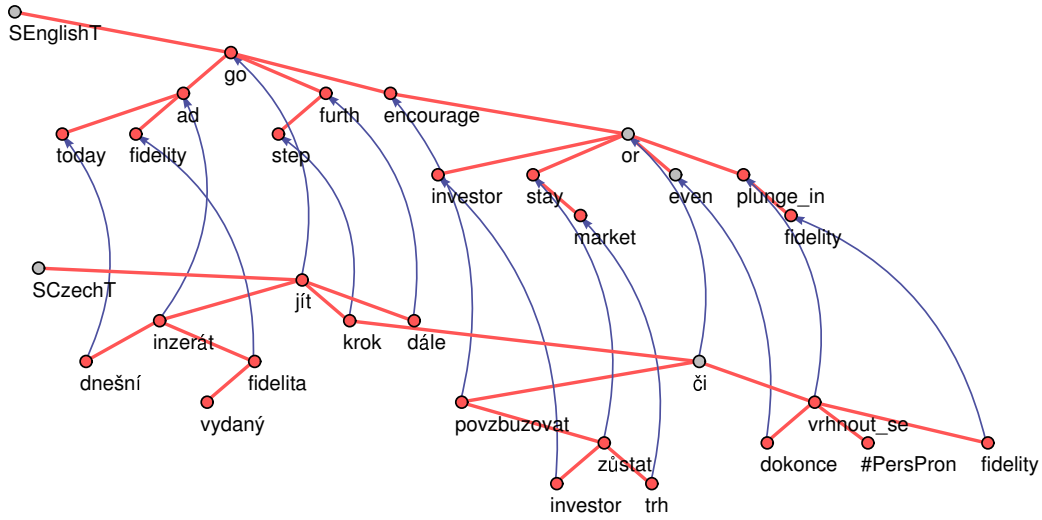


While more than 44 million Americans had sworn off cigarettes by 1990, the percentage of women quitting was slightly lower than that of men.
Do roku 1990 zanechalo kouření více než 44 milionů Američanů, procento žen mezi nimi však bylo poněkud nižší než u mužů.

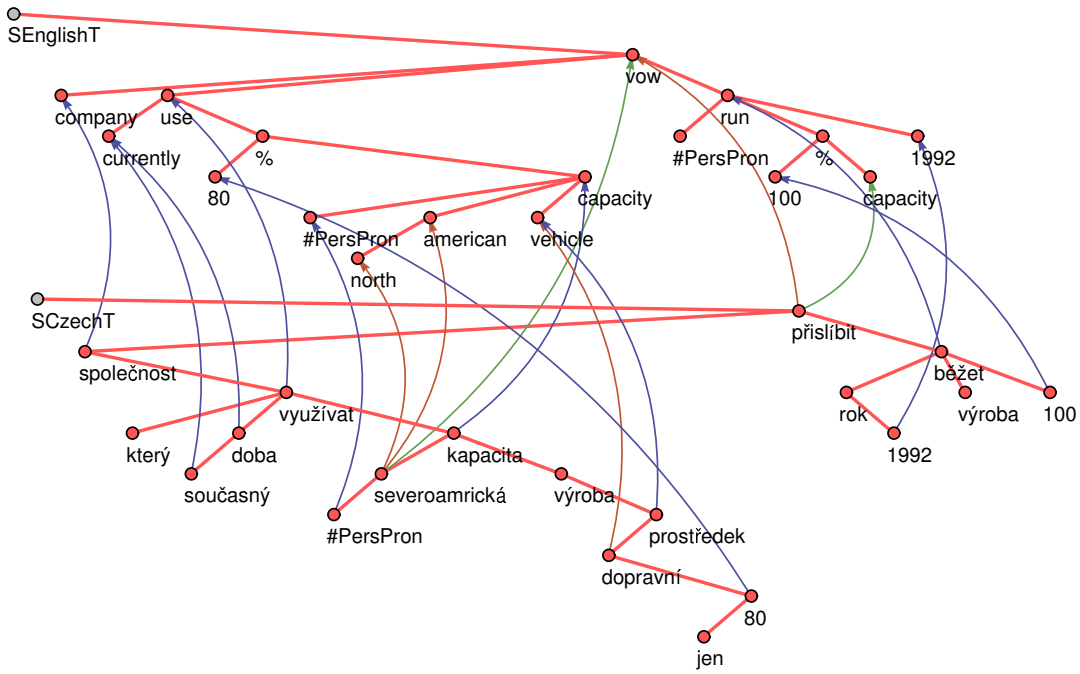
B.4 Sentences from PCEDT



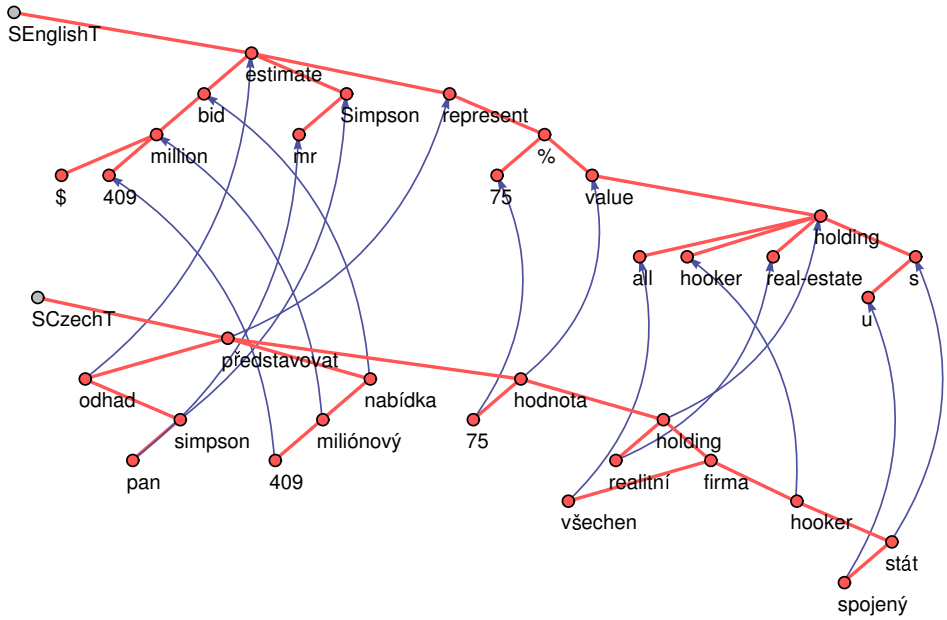
Instead of being denounced as an evil agent of imperialism, Radio Free Europe is more likely to draw the criticism that its programs are too tame, even boring.
Místo aby bylo označováno jako ďábelský agent imperialismu, Rádio Svobodná Evropa bude spíše kritizováno za to, že jeho programy jsou příliš krotké, nebo dokonce nudné.



Today's Fidelity ad goes a step further, encouraging investors to stay in the market or even to plunge in with Fidelity.
Dnešní inzerát vydaný Fidelity jde o krok dále, povzbuzuje investory zůstat na trhu či dokonce vrhnout se na něj spolu s Fidelity.

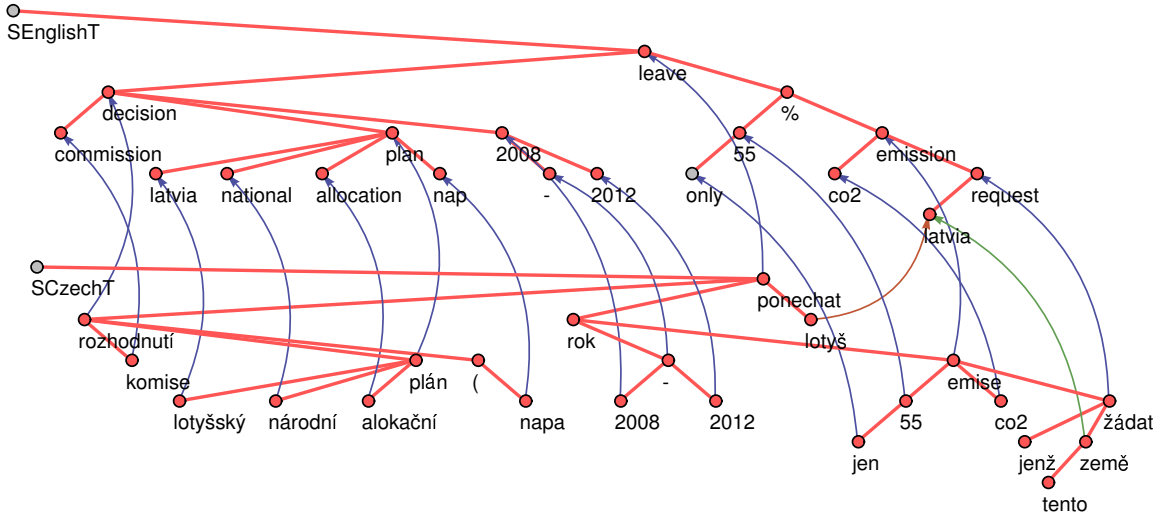


The company, currently using about 80 % of its North American vehicle capacity, has vowed it will run at 100 % of capacity by 1992.
Společnost, která v současné době využívá svou severoamrickou kapacitu na výrobu dopravních prostředků jen z 80 %, přislíbila, že v roce 1992 poběží výroba na 100 %.

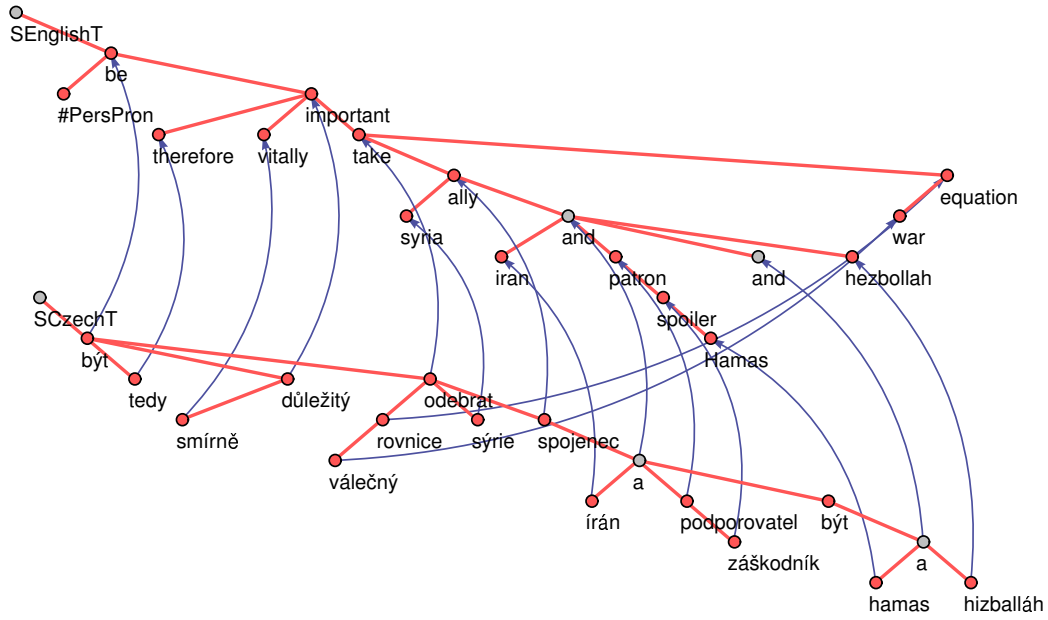


The $ 409 million bid is estimated by Mr. Simpson as representing 75 % of the value of all Hooker real-estate holdings in the U. S.
Podle odhadu pana Simpsona představuje 409 milionová nabídka 75 % hodnoty všech realitních holdingů firmy Hooker ve Spojených státech.

B.5 Sentences from Project Syndicate (Named Entities)

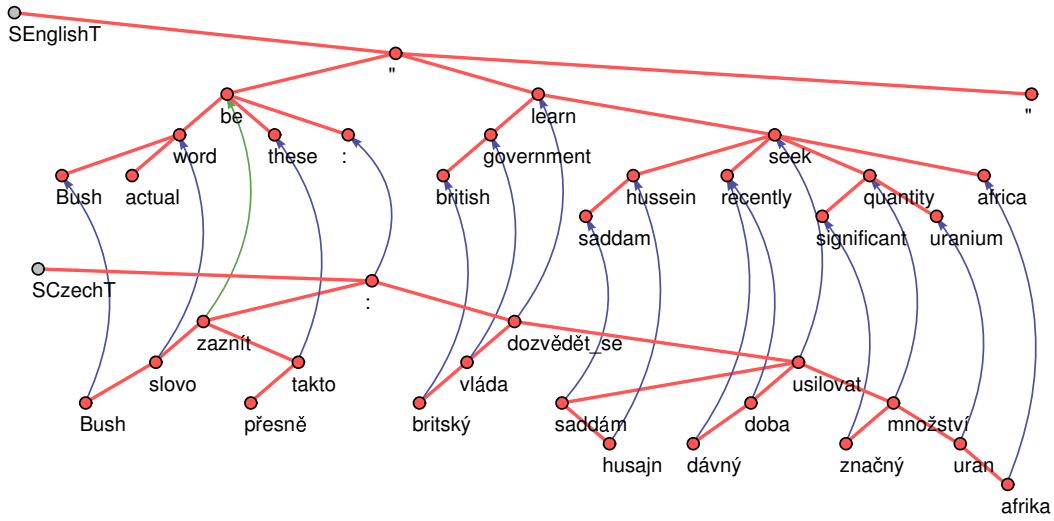


The Commission's decision on Latvia's National Allocation Plan (NAP) for 2008 – 2012 left only 55 % of the CO2 emissions that Latvia requested.
Rozhodnutí komise o lotyšském Národním alokačním plánu (NAP) pro roky 2008 – 2012 ponechalo Lotyšku jen 55 % emisí CO2, o něž tato země žádala.



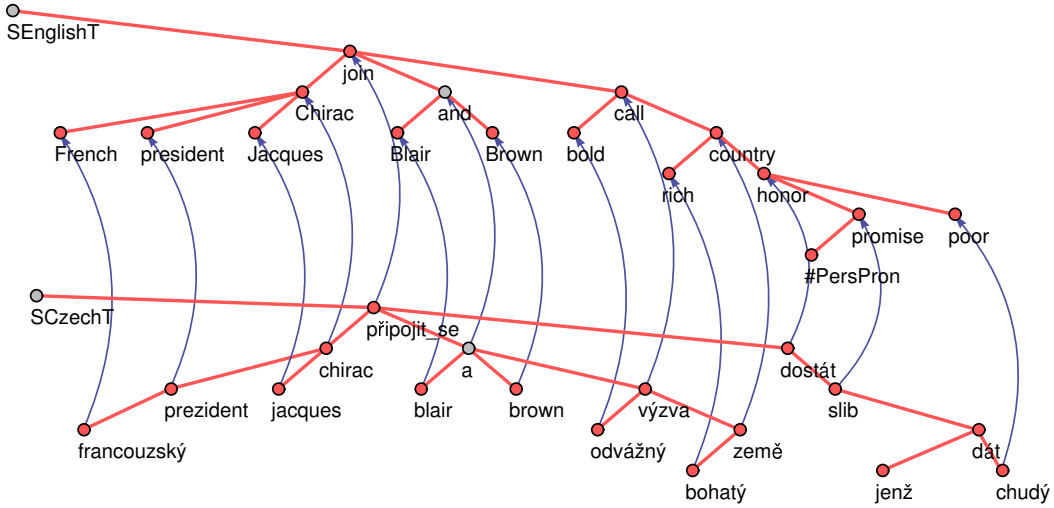
It is therefore vitally important to take Syria, an ally of Iran and the patron of spoilers such as Hamas and Hezbollah, out of the war equation.

Je tedy nesmírně důležité z válečné rovnice odebrat Sýrii, spojence Íránu a podporovatele záškodníků, jako jsou Hamás a Hizballáh.



Bush's actual words were these: "The British government has learned that Saddam Hussein recently sought significant quantities of uranium from Africa."

Bushova slova zazněla přesně takto: "Britská vláda se dozvěděla, že Saddám Husajn v nedávné době usiloval o značné množství uranu z Afriky."



French President Jacques Chirac has joined Blair and Brown in a bold call for rich countries to honor their promises to the poor.
Francouzský prezident Jacques Chirac se připojil k Blairovi a Brownovi v odvážné výzvě k bohatým zemím, aby dostály slibům, jež daly chudým.

Appendix C

TectoMT Blocks Used for Tectogrammatical Alignment

TectoMT blocks that were created within the scope of this thesis will be introduced first. There are two blocks for GIZA++ t-tree alignment, one block for transferring word alignment into t-tree alignment, the t-aligner itself consists of two blocks, and the last block concerns t-alignment evaluation.

Print::Tlemma_bitexts
Extracts t-lemmas from tectogrammatical trees and prints it to the standard output in format: `<sentence_id><TAB><english_tlemmas><TAB><czech_tlemmas>` Each bundle generates one line.

Align_SEnglishT_SCzechT::Giza_alignment
Reads the alignment file generated by GIZA++ and copies the alignment into TMT files.

Align_SEnglishT_SCzechT::Walign_to_Talign
If there exist any alignment on the word layer in TMT file, this block transfers it to the tectogrammatical layer.

Align_SEnglishT_SCzechT::Greedy_1_to_1_alignment
The first part of tectogrammatical aligner. Greedy feature-based algorithm which generates 1:1 alignment only. It uses probabilistic dictionary, usage of GIZA++ alignment is optional.

Align_SEnglishT_SCzechT::Complete_1_to_N_relations
The second phase of the aligner. Other connections are added.

Eval::T_alignment_evaluation
Evaluates the alignments made by both t-aligner and GIZA++ tool. Shows results for all three evaluation variants.

There is the list of blocks previously existing in TectoMT that were used for Czech and English tectogrammatical analysis. We were using SVN revision 600.

```
SEnglishW_to_SEnglishM::Penn_style_tokenization
SEnglishW_to_SEnglishM::TagTnT
SEnglishW_to_SEnglishM::Fix_mtags
SEnglishW_to_SEnglishM::Lemmatize_mtree
SEnglishM_to_SEnglishP::Phrase_parsing
SEnglishP_to_SEnglishA::Mark_heads
SEnglishP_to_SEnglishA::Build_atree
SEnglishP_to_SEnglishA::Rehang_appos
SEnglishP_to_SEnglishA::Fix_topology
SEnglishP_to_SEnglishA::Fix_multiword_prep_and_conj
SEnglishP_to_SEnglishA::Assign_coap_afuns
SEnglishA_to_SEnglishT::Mark_auxiliary_nodes
SEnglishA_to_SEnglishT::Build_ttree
SEnglishA_to_SEnglishT::Fill_is_member
SEnglishA_to_SEnglishT::Fix_tlemmas
SEnglishA_to_SEnglishT::Assign_coap_functors
SEnglishA_to_SEnglishT::Distrib_coord_aux
SEnglishA_to_SEnglishT::Mark_clause_heads
SEnglishA_to_SEnglishT::Mark_passives
SEnglishA_to_SEnglishT::Assign_functors
SEnglishA_to_SEnglishT::Mark_infin
SEnglishA_to_SEnglishT::Mark_dsp_root
SEnglishA_to_SEnglishT::Mark_parentheses
SEnglishA_to_SEnglishT::Recompute_deepord
SEnglishA_to_SEnglishT::Assign_nodetype
SEnglishA_to_SEnglishT::Assign_sempos
SEnglishA_to_SEnglishT::Assign_grammatemes
SEnglishA_to_SEnglishT::Detect_formeme
SEnglishA_to_SEnglishT::Detect_voice
SEnglishA_to_SEnglishT::Mark_person_names

SCzechW_to_SCzechM::Tokenize
SCzechW_to_SCzechM::Analyze_and_tag.pm
SCzechM_to_SCzechA::McD_parser_local
SCzechM_to_SCzechA::Fix_atree_after_McD
SCzechA_to_SCzechT::Mark_auxiliary_nodes
SCzechA_to_SCzechT::Build_ttree
SCzechA_to_SCzechT::Rehang_unary_coord_conj
SCzechA_to_SCzechT::Fill_is_member
SCzechA_to_SCzechT::Assign_coap_functors
SCzechA_to_SCzechT::Distrib_coord_aux
SCzechA_to_SCzechT::Mark_clause_heads
SCzechA_to_SCzechT::Mark_relclause_heads
SCzechA_to_SCzechT::Fix_tlemmas
SCzechA_to_SCzechT::Recompute_deepord
SCzechA_to_SCzechT::Assign_nodetype
SCzechA_to_SCzechT::Assign_grammatemes
SCzechA_to_SCzechT::Detect_formeme
```