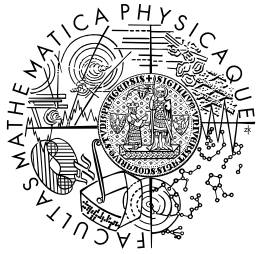

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Adéla Limbursk á

Semantic information from FrameNet and the possibility of its transfer to Czech data

Institute of Formal and Applied Linguistics

Supervisor of the master thesis: doc. RNDr. Markéta Lopatková, Ph.D.
Study programme: Informatics
Study branch: Mathematical Linguistics

Prague 2016

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ..... signature of the author

Title: Semantic information from FrameNet and the possibility of its transfer to Czech data

Author: Adéla Limbursk á

Institute: Institute of Formal and Applied Linguistics

Supervisor: doc. RNDr. Markéta Lopatková, Ph.D., Institute of Formal and Applied Linguistics

Abstract: The thesis focuses on transferring FrameNet annotation from English to Czech and the possibilities of using the resulting data for automatic frame prediction in Czech. The first part, annotation transfer, has been performed in two ways. First, a parallel corpus of English sentences and their human created Czech translations (PCEDT) was used. Second, a much larger parallel corpus was created using machine translation of FrameNet example sentences. This corpus was then used to transfer the annotation as well. The resulting data were partially evaluated and some of the automatically detectable errors were filtered out. Subsequently, the data were used as an input for two machine learning methods, decision trees and support vector machines. Since neither of the machine learning experiments brought impressive results, further manual correction of the data annotation was performed, which helped increase the accuracy of the prediction. However, as the accuracy reported in related papers is notably higher, the thesis also discusses different approaches to feature selection and the possibility of further improvement of the prediction results using these methods.

Keywords: FrameNet frame semantics machine learning word sense disambiguation Czech

First, I would like to thank my family for providing me with support and encouragement throughout my studies.

I am also grateful to my supervisor, doc. RNDr. Markéta Lopatková, Ph.D., for the guidance and advice she gave me during the process of writing the thesis.

Furthermore, I would like to thank Mgr. Rudolf Rosa for his help with machine translation and automatic alignment of the data, and Mgr. Barbora Vidová Hladká, Ph.D. for her suggestions regarding machine learning methods.

Contents

Introduction

In the present thesis, we describe experiments aimed at an automatic assignment of semantic information, namely the FrameNet semantic frames, to words in Czech sentences.

First, the underlying theory is described. The annotation we attempt to automatically create is based on the theory of frame semantics which was introduced by Charles Fillmore at the end of 1970s. The key notion of the theory is a *frame*, which covers all ideas, images and mental concepts that a speaker associates with a certain word or phrase of a language. Frames, as Fillmore [1976] suggests, can be used to describe meanings of phrases and sentences in any language.

In 1997, a project named FrameNet started at the University of California in Berkeley, the aim of which was to create a frame-based on-line dictionary. The resulting database currently contains more than 1,000 thoroughly described lexical units with example sentences, full-text frame annotation and a complex system of frame-to-frame relations.

Using the annotation from both example sentences [FrameNet Index of Lexical Units] and the full-text section [Full Text Index], we managed to transfer the FrameNet information to Czech. For the full-text annotation transfer, we used the Prague Czech-English Dependency Treebank (PCEDT) [Hajič et al., 2012], a parallel treebank of English sentences and their Czech translations created by human translators. It contains dependency trees for all sentences on both sides. To exploit the information in the example sentences with the FrameNet annotation, we translated them into Czech using the TectoMT machine translation system [Popel and Žabokrtský, 2010] which also generated dependency trees in the same format which is used by the PCEDT. Thus, we obtained a reasonably large collection of Czech frame-annotated and dependency-parsed data. A subsequent evaluation showed that the quality of both transfer methods is comparably high: the full-text data were evaluated as 79% correct and the machine translated data reached a 70% success rate.

The obtained data were then used as an input for two different machine learning tools: one based on decision trees and second based on support vector machines. Using both of these methods, we attempted to predict the correct frame for the 100 most frequent lemmas in the dataset. First, we carried out a baseline experiment, in which we found the most frequent frame for each lemma in the training data and assigned it to every instance of the lemma in the test data. This method had a 60.3% average accuracy[1] in a 10-fold cross validation. After performing machine

[1]By *accuracy* we mean the count of correctly annotated tokens divided by the count of all tokens.

learning experiments on different sets of features obtained from the a-layer of the dependency trees, our best results were a 62.2% accuracy using the decision trees (71.2% when the unannotated results were included) and a 62.6% accuracy using the support vector machines (71.4% with the unannotated results).

Furthermore, we corrected the data manually which helped improve the results to a final 68.3% accuracy on annotated tokens and 76.1% accuracy on all tokens.

1. Frame Semantics and FrameNet

1.1 Frame Semantics Theory

Frame semantics is a linguistic theory developed by Charles J. Fillmore in the end of 1970s. It attempts to systematically describe the concepts that a speaker of a language needs to understand in order to be able to interpret a certain expression of that language. Due to the fact that the main focus of the description is on the mental structures of a speaker and not on particular language constructions, it can be applied (without major changes) on any language. [Fillmore, 1982]

Frame semantics was developed by expanding and modifying Fillmore's earlier case grammar theory, which was based on some generative grammar assumptions, such as the core position of syntax in language description or the belief that for all languages, there is one set of syntactic relations and languages only differ in their surface realisation. On the other hand, case grammar focused mainly on semantics, in which it differed from generative grammar.

The *case* in the case grammar is a way of expressing the relation between a noun phrase and the predicate. "The case notions comprise a set of universal, presumably innate, concepts which identify certain types of judgments which human beings are capable of making on the events that are going on around them, judgments on such matters as who did it, who it happened to, what got changed etc." [Fillmore, 1968, p. 46] Cases are a part of the deep structure of a sentence and therefore are not influenced by surface syntactic relations. The following example shows this:

1. John (Agentive – A) opened the door (Objective – O)
2. John (A) opened the door (O) with the key (Instrument – I)
3. the key (I) opened the door (O)
4. the door (O) opened

(for further information about the case grammar, see Fillmore [1968])

In the 1970s, reacting both to critical responses to the case grammar and his own further research, Fillmore began to modify the case grammar theory. Influenced by the findings of cognitive linguistics, Fillmore [1976, p. 23] stated:

> "My effort is to look for what can be known about the workings of language through a consideration of the processes of communication. A proposal that I favor is that in characterizing a language system we must add to the description of grammar and lexicon a description of the congitive and interactional 'frames' in terms of which the language-user interprets his environment, formulates his own messages, understands the messages of others, and accumulates or creates an internal model of his world."

This turn resulted in formulating a new theory that was named frame semantics and was based on a belief that lexical and encyclopedic knowledge of a speaker of a language are connected.

The core notion of the frame semantics is a *frame* (or schema, script, scenario, ideational scaffolding, cognitive model etc.), which denotes a set of images and experiences that together form the mental structure that is necessary for a speaker to understand an expression in a language. Various components of this set, such as participants of the action expressed by a word, its circumstances, presuppositions or the speaker's attitude, are called *frame elements*. There are two types of frame elements: *core* ones, which have to be conceptually present for a statement to make sense, and *peripheral* ones, which do not necessarily have to be present. Peripheral elements usually include information about time, place and other circumstances of an action.

Frame elements can be thought of as a modification of deep cases in that they represent the way certain words of an utterance are related to their governing words and serve to describe how speakers interpret a language. However, one of the most important differences is that while in the case grammar only one limited set of cases exists for all governing words, in frame semantics every frame has its particular set of frame elements.

As an example, Fillmore [1982, p. 116] analyses the verbs *accuse* and *criticize* and distinguishes the following frame elements: "a person who formed or expressed some sort of judgment on the worth or behavior of some situation or individual (and I called such a person the Judge); a person concerning whose behavior or character it was relevant for the Judge to make a judgment (I called this person the Defendant); and some situation concerning which it seemed relevant for the Judge to be making a Judgment (and this I called simply the Situation)." The verb *accuse* can then be described as a verb "usable for asserting that the Judge, presupposing the badness of the situation, claimed that the Defendant was responsible for the Situation". On the other hand, the verb *criticize* is "usable for asserting that the Judge, presupposing the Defendant's responsibility for the Situation, presented arguments for believing that the Situation was in some way blameworthy."

As the example shows, one frame can be evoked by various words, which cor-

responds to the fact that one situation can be viewed from various perspectives. For example, a commercial transaction can be viewed from the perspective of the buyer (in this case, the verb *buy* is used), of the seller (the verb *sell*), of the goods (the verb *cost*) etc. However, frames are not only evoked by verbs but also by other words, such as nouns that denote an action (e.g. *replacement*), relational nouns (*brother*), artifact nouns (*house*), some adjectives (e.g. ones that describe emotions such as *happy*), adverbs (similarly as in the case of adjectives, one of the prominent types of frame-evoking adverbs are ones that describe emotions, e.g. *happily*) and even some prepositions (*in*).

In a broader sense, the term frame can also be used to denote a set of properties of the communication situation in which an utterance is used (this type of frame is sometimes called the *interactional frame*). This is most notable in situations associated with specific language constructions. Knowing that a certain text is a contract, an obituary or a marriage proposal can help us understand utterances that are specific for these contexts and would be more difficult to understand without knowing the interactional frame. This knowledge can also help us predict how a particular type of text will be structured. [Fillmore, 1982, p. 117]

Frames can be used to describe the understanding of evaluative adjectives, such as the adjective *good*. To understand phrases like 'a good pen', 'good coffee', 'a good pilot', 'a good mother' etc., we need to know what part a pen, coffee, a pilot or a mother can play in a human life and in what way each of these entities can be good. Similarly, to understand the word *imitation* as used in the phrase 'imitation coffee', the speaker needs to understand the purpose of coffee and the reason why anyone would create an imitation of it. [Fillmore, 1982, p. 133]

1.2 The FrameNet Database

In 1997, a project named FrameNet started at the University of California in Berkeley, the aim of which was to create a frame-based on-line dictionary as suggested by Fillmore and Atkins [1992]. The database, which has been gradually expanded ever since, has the following sections, as described on the FrameNet website[1] [FrameNet Data]:

> "**Full Text Annotation:** In addition to our ongoing lexicographic work, FrameNet has begun to annotate some continuous texts, as a demonstration of how frame semantics can contribute to text understanding. This style of annotation typically involves marking frame elements of frames evoked by multiple predicators in each sentence or even in each clause.

[1] https://framenet.icsi.berkeley.edu/fndrupal/framenet_data, accessed 05–26–2016

Frame Index: Frame definitions, semantic roles/frame elements (FEs), and other frame information.

Lexical Unit Index: Word senses (Lexical Units) with annotation and related syntactic patterns report.

FrameSQL: Search the FrameNet data using the FrameSQL tool (Prof. Sato Hiroaki).

FrameGrapher: Interact with a visual representation of the frame-frame relations in the FrameNet data."

The information that is most necessary for our task of frame prediction is the annotation of sentences, whether they be examples of lexical units or full text annotated data. On the FrameNet website[2] [About FrameNet], the annotation is described as follows:

> "Formally, FrameNet annotations are sets of triples that represent the FE realizations for each annotated sentence, each consisting of a frame element name (for example, Food), a grammatical function (say, Object) and a phrase type (say, noun phrase (NP)). [...] The downloadable XML version of the data includes these three layers (and several more not discussed here) for all of the annotated sentences, along with complete frame and FE descriptions, frame-frame relations, and lexical entries for each annotated LU. Most of the annotations are of separate sentences annotated for only one LU, but there are also a collection of texts in which all the frame-evoking words have been annotated; the overlapping frames provide a rich representation of much of the meaning of the entire text. The FrameNet team have defined more than 1,000 semantic frames and have linked them together by a system of frame relations, which relate more general frames to more specific ones and provide a basis for reasoning about events and intentional actions."

Currently, the freely downloadable data contain almost 200,000 example sentences for the total of about 10,000 annotated lexical units (for further information about the data, see Section 3.1). The full text annotation is comprised of about 100 articles of different kinds, but only 6 of these, namely the Wall Street Journal Texts from the PropBank Project, were suitable for our task (see Section 2.1).

[2]https://framenet.icsi.berkeley.edu/fndrupal/about, accessed 05–26–2016

2. Using Parallel Data to Transfer FrameNet Annotation from English to Czech

As the first step towards an automatic FrameNet annotation for Czech, it was necessary to collect sufficient Czech data containing the information about semantic frames. Our first attempt was to transfer the information from the FrameNet full-text annotated data to Czech using a parallel treebank.

2.1 FrameNet Full Text Annotation

Among other data, the FrameNet database contains texts in which all frame evoking lexical units and their frame elements have been annotated. A small portion of these texts are the Wall Street Journal Texts from the PropBank Project [Palmer et al., 2005]. Since the English part of the Prague Czech-English Dependency Treebank (PCEDT) [Hajič et al., 2012] is wholly comprised of the Penn Treebank [Marcus et al., 1993] – Wall Street Journal Section texts,[1] we have chosen these FrameNet full-text-annotated data as the first and most easily transferable source of FrameNet annotation.

The Wall Street Journal portion of the full text annotated data contains 6 articles on different topics. In these articles, there are 337 sentences, in which 418 different frames are annotated (2,121 annotated frames in total). That means there are on average about 5.1 occurrences of each frame and about 6.3 frame annotations per sentence.

The data can be downloaded in an XML format which contains the sentences and their annotation on various layers. As an example, we show some parts of the annotation of the sentence "I have seen one or two men die, bless them." (Listing 2.1). This sentence is taken from one of the articles and has been chosen mainly because it is relatively short but contains more than one frame annotation.

First, as can be seen in the excerpt from the code below, the text of the sentence and the information about its surface structure are provided. Every label (in this case, the Penn Treebank style annotation and some further information that was not used in our task) is assigned to a certain word or phrase by specifying the position of the beginning and the end of the desired segment in the whole sentence.

[1] The articles contained in the Wall Street Journal section of the PropBank Project are a subset of the Wall Street Journal texts included in the Penn Treebank.

```
1  <sentence corpID="115" docID="23009" sentNo="3" paragNo="26" aPos=
       "0" ID="2824265">
2      <text>I have seen one or two men die , bless them .</text>
3      <annotationSet cDate="02/05/2004 08:29:41 PST Thu" status="
           UNANN" ID="4528260">
4          <layer rank="1" name="PENN">
5              <label end="0" start="0" name="PP" />
6              <label end="5" start="2" name="VHP" />
7              <label end="10" start="7" name="VVN" />
8              <label end="14" start="12" name="cd" />
9              <label end="17" start="16" name="cc" />
10             <label end="21" start="19" name="cd" />
11             <label end="25" start="23" name="nns" />
12             <label end="29" start="27" name="VV" />
13             <label end="31" start="31" name="," />
14             <label end="37" start="33" name="VV" />
15             <label end="42" start="39" name="PP" />
16             <label end="44" start="44" name="sent" />
17         </layer>
18         <layer rank="1" name="NER" />
19         <layer rank="1" name="WSL">
20             <label end="5" start="2" name="NT" />
21             <label end="31" start="31" name="NT" />
22             <label end="44" start="44" name="NT" />
23             <label end="0" start="0" name="NT" />
24             <label end="17" start="16" name="NT" />
25             <label end="42" start="39" name="NT" />
26         </layer>
27     </annotationSet>
```

Listing 2.1: Text and surface syntax in FrameNet XML

In the next part of the annotation, the information about frames is provided. In most sentences, more than one frame is annotated and in such case, there is a different annotation set for every frame. Every annotation set contains information about the frame, such as its name and ID, and several layers of annotation. For our task, the most important one is the Target layer. Every frame annotated in a sentence has a target, also called frame evoking element, that is a word or phrase that evokes the given frame. In the example below (Listing 2.2, line 5), the first annotated frame, Perception_experience (line 1), is evoked by the target that consists of the characters on the 7^{th} to 10^{th} position in the sentence (starting from 0). In the Listing 2.1 above, we can see that it is the word *seen*. Similarly, the second frame, Death (line 21), is evoked by the verb *die* (lines 22–24).

```
<annotationSet cDate="07/08/2004 04:42:20 PDT Thu" luID="1535"
    luName="see.v" frameID="70" frameName="Perception_experience"
    status="MANUAL" ID="4529010">
    <layer rank="1" name="Target">
        <label cBy="MiL" end="10" start="7" name="Target" />
    </layer>
    <layer rank="1" name="FE">
        <label cBy="MiL" feID="331" bgColor="FF0000" fgColor="
            FFFFFF" end="0" start="0" name="Perceiver_passive" />
        <label cBy="MiL" feID="332" bgColor="0000FF" fgColor="
            FFFFFF" end="29" start="12" name="Phenomenon" />
    </layer>
    <layer rank="1" name="GF">
        <label end="0" start="0" name="Ext" />
        <label end="29" start="12" name="Obj" />
    </layer>
    <layer rank="1" name="PT">
        <label end="0" start="0" name="NP" />
        <label end="29" start="12" name="NP" />
    </layer>
    <layer rank="1" name="Other" />
    <layer rank="1" name="Sent" />
    <layer rank="1" name="Verb" />
</annotationSet>
<annotationSet cDate="07/08/2004 04:43:30 PDT Thu" luID="924"
    luName="die.v" frameID="53" frameName="Death" status="MANUAL"
    ID="4529011">
    <layer rank="1" name="Target">
        <label cBy="MiL" end="29" start="27" name="Target" />
    </layer>
    <layer rank="1" name="FE">
        <label cBy="MiL" feID="218" bgColor="FF0000" fgColor="
            FFFFFF" end="25" start="12" name="Protagonist" />
    </layer>
    <layer rank="1" name="GF">
        <label end="25" start="12" name="Ext" />
    </layer>
    <layer rank="1" name="PT">
        <label end="25" start="12" name="NP" />
    </layer>
    <layer rank="1" name="Other" />
    <layer rank="1" name="Sent" />
    <layer rank="1" name="Verb" />
</annotationSet>
```

Listing 2.2: Full text annotation in FrameNet XML

Another interesting layer is the one named FE, which captures the elements of each frame. For the Perception_experience frame, two frame elements are present in the sentence: Perceiver_passive (line 6), which is realized by the word on positions 0–0 (that is *I*), and Phenomenon (line 7), which corresponds to the phrase on positions 12–29 (that is *one or two men die*).

The sentence contains various other frame evoking elements, namely *men*, which evokes the frame called People, and *one* and *two*, which both evoke the Cardinal_numbers frame. Each of the frames has its frame elements that are annotated in the way shown above.

From the XML files, we were able to easily extract the information about frames for all the sentences. To transfer the gathered information into Czech, we used a parallel treebank of English sentences and their Czech translations, PCEDT 2.0.

2.2 PCEDT 2.0

The Prague Czech-English Dependency Treebank 2.0 (PCEDT) [Hajič et al., 2012] is an automatically aligned parallel treebank of English texts and their Czech translations. "The English part contains the entire Penn Treebank – Wall Street Journal Section (LDC99T42). The Czech part consists of Czech translations of all of the Penn Treebank-WSJ texts. The corpus is 1:1 sentence-aligned. An additional automatic alignment on the node level (different for each annotation layer) is part of this release, too."[2] [Introduction to PCEDT]

In the treebank, each sentence is annotated on three different layers (plus on an additional p-layer for the Penn-Treebank-style annotation of the English part of the treebank) that capture various properties of the sentence. In addition to the layers of annotation, the data also contain the w-layer which captures only the raw text of the sentence and does not contain any annotation.

The shallowest annotation layer of the treebank is the *m-layer* (morphological layer). It contains information about morphological properties of individual words and punctuation marks, such as lemma and morphological tag. The tags differ in the English and Czech annotation. The English part uses the Penn Treebank set of tags that contain 2 to 4 characters and capture the part of speech and in some cases also the form of a word. In Czech, the positional morphological tags are used. A Czech positional tag is a sequence of 15 characters, each of which represents a certain morphological feature of the analysed word form.

For example, the word *art* that appears in the English part of the data is tagged NN, that is singular or mass noun. Its Czech counterpart, *umění*, is tagged NNNS1-----A----, which can be read as noun (position 1), general noun (pos. 2),

[2] https://ufal.mff.cuni.cz/pcedt2.0/, accessed 05–19–2016

neuter (pos. 3), singular (pos. 4), nominative (pos. 5), affirmative (i.e. not negative, pos. 11). For further information about the tags, see Hajič [2004].

For our task of frame prediction, we use the lemma and one position of the morphological tag, namely the first character, which represents the general part of speech.

The next layer, *a-layer* (analytical layer), represents the surface syntactic structure of a sentence and contains labels such as predicate, subject, object etc. (this category is called an analytical function). These labels could be of some help for the task of frame prediction, however, it was not possible to use them. The reasons can be found in Chapter 3.

The a-tree structure has a form of a dependency tree. It is therefore possible to easily extract information about the direct dependency relations of a node. This can be useful for the task of word sense disambiguation, which is closely related to the task of frame prediction.

The a-layers of an English sentence and its Czech equivalent are word-to-word aligned. An example of the two aligned a-layer dependency trees is shown in Figure 2.1 which has been obtained from the PCEDT website[3] [Introduction to PCEDT].

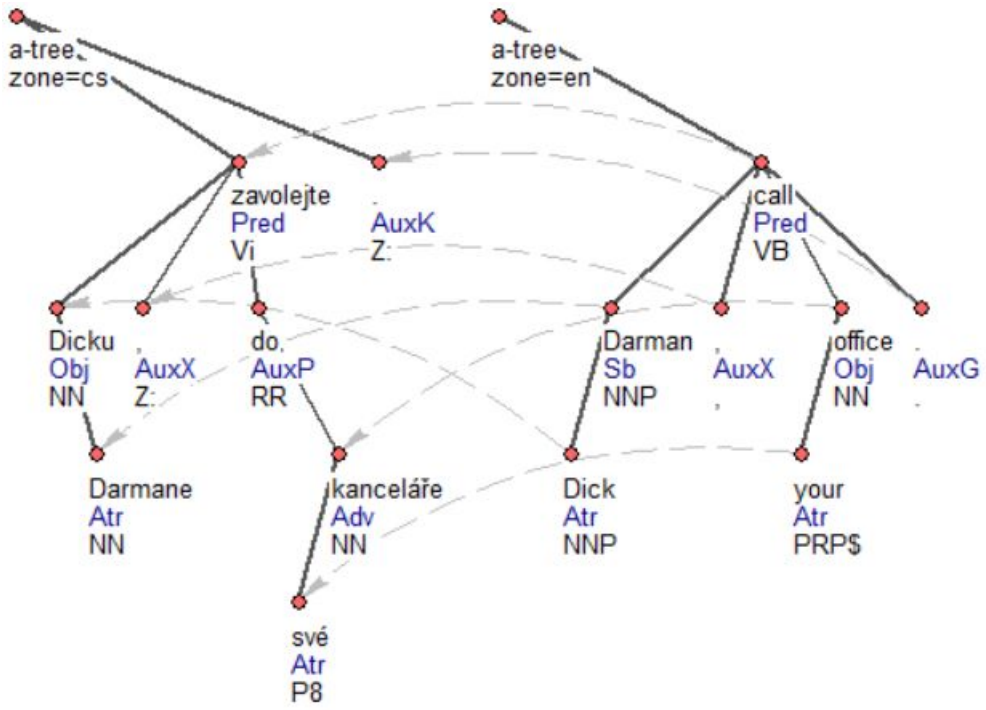

Figure 2.1: The a-layer annotation in the PCEDT

[3]https://ufal.mff.cuni.cz/pcedt2.0/, accessed 05–19–2016

The last layer of the PCEDT annotation is the *t-layer* (tectogrammatical layer) which "is a linguistic representation that combines syntax and, to a certain extent, semantics, in the form of semantic labeling, anaphora resolution and argument structure description based on a valency lexicon"[4] [Introduction to PCEDT]. It is based on the Functional Generative Description (FGD, see for example Sgall [1967] or Sgall et al. [1986]). On this layer, only content words are displayed and each of them is assigned with multiple labels. None of the labels have been used in our task but some could be of great help in a closely related task, the prediction of frame elements of a frame that is given, since the semantic frame and the t-layer structure represent a similar type of information, although in a different manner.

The data are in the treex format (see Žabokrtský [2011]) which somewhat resembles the XML format. In the example below, the analytical representation of the phrase *bless them* (from the example sentence above) is shown. Generally, the annotation starts at the topmost node (that is the node representing the word that is not dependent on any other word of the sentence) and follows through its children until it reaches the leaves of the dependency tree. In our case, therefore, the word *bless* is identified first (line 1). However, right after the identification, the information about its children follows (lines 2–14). That is why the form *them* appears higher in the code than the form *bless*.

```
1  <LM id="EnglishA-wsj_0089-s65-t11">
2    <children id="EnglishA-wsj_0089-s65-t12">
3      <form>them</form>
4      <lemma>them</lemma>
5      <tag>PRP</tag>
6      <no_space_after>1</no_space_after>
7      <ord>11</ord>
8      <afun>Obj</afun>
9      <p_terminal.rf>EnglishP-wsj_0089-s65-t12</p_terminal.rf>
10     <alignment>
11       <counterpart.rf>a-wsj0089-001-p1s64W11-Ta</counterpart.rf>
12       <type>gdfa</type>
13     </alignment>
14   </children>
15   <form>bless</form>
16   <lemma>bless</lemma>
17   <tag>VB</tag>
18   <ord>10</ord>
19   <afun>Adv</afun>
20   <p_terminal.rf>EnglishP-wsj_0089-s65-t11</p_terminal.rf>
21   <alignment>
22     <LM>
23       <counterpart.rf>a-wsj0089-001-p1s64W12-Ta</counterpart.rf>
```

[4] https://ufal.mff.cuni.cz/pcedt2.0/, accessed 05–19–2016

```
24          <type>int.gdfa</type>
25        </LM>
26        <LM>
27          <counterpart.rf>a-wsj0089-001-p1s64W10-Ta</counterpart.rf>
28          <type>gdfa</type>
29        </LM>
30      </alignment>
31    </LM>
```

Listing 2.3: Treex format – English part

For the transfer of information from English FrameNet full text annotation to Czech we used the counterpart.rf (counterpart reference) which links every English word to its Czech counterpart. In the example below (Listing 2.4), the treex representation of the phrase *bůh jim žehnej*, which is the Czech equivalent of the above phrase *bless them*, is shown. Note that the IDs of the Czech words (lines 1, 3 and 10) are the same as the counterpart.rfs above (Listing 2.3, lines 23, 27 and 11).

```
1  <children id="a-wsj0089-001-p1s64W12-Ta">
2    <children>
3      <LM id="a-wsj0089-001-p1s64W10-Ta">
4        <form>bůh</form>
5        <lemma>bůh</lemma>
6        <tag>NNMS1-------A-----</tag>
7        <ord>10</ord>
8        <afun>Sb</afun>
9      </LM>
10     <LM id="a-wsj0089-001-p1s64W11-Ta">
11       <form>jim</form>
12       <lemma>on-1</lemma>
13       <tag>PPXP3--3-------</tag>
14       <ord>11</ord>
15       <afun>Obj</afun>
16     </LM>
17   </children>
18   <form>žehnej</form>
19   <lemma>žehnat_:T</lemma>
20   <tag>Vi-S---2--A-----</tag>
21   <no_space_after>1</no_space_after>
22   <ord>12</ord>
23   <afun>Obj</afun>
24   <is_member>1</is_member>
25 </children>
```

Listing 2.4: Treex format – Czech part

The PCEDT data are the most reliable ones we were able to get. Although both the morphological and syntactic analysis of the sentences and the word-level alignment between the two languages have been obtained automatically, the amount of errors in the alignment is reasonably low, as the evaluation of the frame element transfer has shown (see Section 2.3). The major advantage of the data is that the Czech translations have been created manually by professional translators (who were instructed to keep the translations as close to the original as possible) and therefore can be considered highly adequate.

2.3 Transferring the Information

The transfer of the information (together with an extraction of some interesting features of individual words) has been carried out by a perl script that performs two major steps.

In the first step, information from the PCEDT is extracted. On the English side, we are only interested in the information about word order and alignment: word order helps us find the corresponding English PCEDT phrase for every annotated FrameNet phrase, and alignment information (counterpart.rf in the example in the previous section) lets us identify the Czech translation of each English phrase. On the Czech side, we extract some more information in order to obtain features that will help us in our task of frame prediction. These features include information about tree structure (for each node, tags of its parent and its children are captured), word form, lemma, POS-tag and analytical function, although the latter could not be used in our task (see Chapter 3). We also extract the information about word order, even though we do not use it in our frame prediction task – it only serves to make the output accessible to human readers which is important in the process of evaluation.

In the second step, the FrameNet data are exploited. For each annotated sentence in the FrameNet full text annotation, we find its corresponding sentence in the PCEDT. This can mostly be done in a straightforward manner using the order of the sentence in the file. However, in this approach, we encountered several problems, e.g. with sentence boundaries as they were sometimes different in each resource. When this was a regularly appearing issue (such as a different approach to quotation marks in FrameNet and in PCEDT), it was solved using a regular expression that identified problematic sentences and applied rules to match them correctly. When the difference in sentence boundaries did not seem to be a pattern but rather a specific decision of the annotators (this happened twice in the whole data), individual exceptions were added into the script.

After identifying the corresponding English sentence, each annotated FrameNet phrase was matched with its corresponding phrase in the English part of the

PCEDT. This could mostly be done using word order since a majority of the sentences are exactly the same in FrameNet and in PCEDT and therefore the position of each phrase is the same as well. Those cases, in which sentence boundary issues were encountered, could also be solved using word order – if the word on the expected position in the PCEDT sentence was not the one we found in FrameNet, we would recursively move one position left and right in the sentence and eventually find the word. Sometimes, issues with word boundaries appeared. This only happened systematically around hyphens, dashes, slashes and other punctuation marks and therefore could be wholly solved using regular expressions. For a list of all word boundary problems see Attachment 1.

When the correct English phrase in PCEDT was identified, its Czech translation was found in the counterpart.rf feature, without using any additional information. As the alignment had been obtained automatically, it contains a number of errors that have influenced the quality of our transferred data.

2.4 Evaluation

To get an idea about the quality of the information transfer and the resulting data, we have evaluated a part of the Czech texts enriched by FrameNet annotation. The evaluation has been done manually by just one person and has covered all sentences from one of the six articles. We were looking at the correctness of the frames assigned to Czech words and distinguished the following types of errors:

- Alignment errors: a situation when a frame would be correct for a certain word or phrase in the sentence but is assigned to at least one different (wrong) word because of an error in the alignment

- Language differences: a situation when a frame is assigned to the word or phrase that is correctly aligned with its English counterpart, but due to the differences between Czech and English, the Czech phrase does not evoke the same frame as the English one; this type of error also includes cases when the translation was not literal – this happened mainly in cases when literal translation was not possible

- Error originating in FrameNet: this type of error was used to describe the rare situations when we could not agree with the frame that was assigned to the Czech phrase although it was transferred correctly, and found out we also were not satisfied with the one that was assigned to its English counterpart

To illustrate the first error type, we show a part of an English sentence and its Czech translation (with English equivalents of each word), in which a wrong

word is annotated as the frame evoking element. The frame evoking word is highlighted in boldface and the evoked frame is written right after the frame evoking element in brackets. In the correct English annotation (example 1), the frame Temporal_pattern is evoked by the word *rhythmically* and this word is correctly labelled as the frame evoking element. However, in example 2, the same frame is annotated but the word *přivázaná* meaning 'attached' is annotated as the frame evoking element.

(1) *pull* ***rhythmically****(Temporal_pattern) on ropes attached to the same five bells*

(2) *rytmicky* *tahá za lana* ***přivázaná****(Temporal_pattern) ke stejné pětici*
rhythmically pull on ropes **attached**(Temporal_pattern) to same five
zvonů
bells

As can be seen below in the first excerpt from the PCEDT data (Listing 2.5), the word *rhythmically* has an annotated counterpart, the id of which is a-wsj0089-001-p1s5W12-Ta (line 9). However, in the second excerpt, Listing 2.6, which was taken from the Czech part of the data, we can see that the id a-wsj0089-001-p1s5W12-Ta belongs to the word *přivázaná* which means 'attached' (lines 1 and 5).

```
1  <LM id="EnglishA-wsj_0089-s6-t10">
2    <form>rhythmically</form>
3    <lemma>rhythmically</lemma>
4    <tag>RB</tag>
5    <ord>10</ord>
6    <afun>Adv</afun>
7    <p_terminal.rf>EnglishP-wsj_0089-s6-t10</p_terminal.rf>
8    <alignment>
9      <counterpart.rf>a-wsj0089-001-p1s5W12-Ta</counterpart.rf>
10     <type>gdfa</type>
11   </alignment>
12 </LM>
```

Listing 2.5: Example of a wrong alignment – English part

```
<children id="a-wsj0089-001-p1s5W12-Ta">
  <children id="a-wsj0089-001-p1s5W13-Ta">
    [...]
  </children>
  <form>přivázaná</form>
  <lemma>přivázaný_^(*2t)</lemma>
  <tag>AANP4----1A----</tag>
  <ord>12</ord>
  <afun>Atr</afun>
</children>
```

Listing 2.6: Example of a wrong alignment – Czech part

The errors resulting from language differences and impossibility of literal translation can be illustrated using the beginning of the sentence "Mr. Baldwin is also attacking the greater problem: lack of ringers." In the FrameNet annotation, the verb *attacking* is annotated as the frame evoking element for the frame Attack. However, in Czech it is not possible to describe the situation, in which a person points out to a certain problem, using any verb that would be semantically related to the verb *attack*. Instead, a different verb is used which does not evoke the Attack frame.

(3) *Mr. Baldwin is also* ***attacking****(Attack) the greater problem*

(4) *Baldwin rovněž* ***poukazuje****(Attack) na větší problém*
Baldwin also **points out**(Attack) to bigger problem

Several cases also appeared when language differences were the cause of some inconvenient annotation although the result had to be marked as correct with respect to the nature of Czech. A most prominent example of this is the auxiliary verb *být* 'to be', which is a part of many different verbal forms (1^{st} and 2^{nd} person of both singular and plural past tense, all forms of present and past conditional, all forms of future tense of imperfective verbs etc.) Since the auxiliary verb *být* is a part of the complex verb form, it is correctly annotated in the same way as the autosemantic part of the verbal form and we do not mark these situations as errors. However, the verb *být* is thus associated with many frames that have no relation to its meaning (for an example and further information about this phenomenon, see Section 4.1).

To illustrate the type of error that originates in FrameNet data, we will use the previously seen sentence "I have seen one or two men die, bless them." In this sentence, the word *bless* is not annotated as any frame evoking element, even though in the FrameNet lexical unit index, *bless* is associated with the frame called Rite.

The results of the evaluation can be seen in Table 2.1. The numbers in the second column represent the counts of sentence-frame pairs (since for most sentences multiple frames were annotated). A sentence-frame pair was considered correct only in the case when every word in the sentence was either correctly annotated as the frame evoking element of the given frame or correctly annotated as not being one. Note that we are completely leaving out the frame elements, since they are not important for our task. In the third column, the results are expressed in percent.

	Count	Percent
Correct	410	79
Alignment	80	16
Language differences	17	3
FrameNet errors	1	0
Multiple error types	8	2
Sum	516	100

Table 2.1: Evaluation results – parallel data transfer

3. Using Machine Translation to Acquire Further Data

Using the parallel treebank, we obtained fairly reliable data which were, however, insufficient in size. Therefore, we attempted to expand our resources of FrameNet annotation for Czech using machine translation.

3.1 FrameNet Example Annotation

As our data resource for the English FrameNet annotation, we chose the sentences that appear as examples for individual lexical units (LUs) in the FrameNet database. The freely downloadable data on the FrameNet website [FrameNet Index of Lexical Units] currently (as of June 2016) contain 13,530 lexical units, out of which 10,201 are accompanied with an annotation (that is, example sentences are provided).

These data differ from the full text annotated data in that there is usually only one frame annotated for each sentence – the frame that is evoked by the lexical unit exemplified with the sentence. Some sentences appear as examples in more than one LU annotation but this is not a frequent case: there are 197,592 annotated example sentences in total (this is also the number of sentence-frame pairs) and 171,008 of them are different, which means there is on average about 1.16 frame annotated per sentence (the full text annotation presented in the previous chapter had on average 6.3 frames per sentence). However, since each occurrence of a particular sentence is as an example of a different lexical unit, the occurrences do not appear together in the data. Thus, we treat them as if they were different sentences.

The data are in an XML format very similar to the one we have seen in Section 2.1. The only difference is that the sentences are not arranged as a text but as examples of individual lexical units. For each lexical unit, the frame it evokes is specified first and then, all example sentences are annotated without specifying the frame again.

The description of one of the lexical units associated with the verb *miss* and the frame it evokes can be seen in Listing 3.1 below. First, the LU information is provided. Among other properties, the name and id of the lexical unit is given as well as the name of the frame it evokes (line 1). Then, the corpora from which the example sentences have been obtained are listed. Next, all elements of the frame are specified (lines 7–14); in the example below, some have been left out. Lastly, the definition of the word and the description of the lexeme (that is, the lexical

unit without distinction of its individual meanings) is provided (line 16).

```
1  <lexUnit status="Created" POS="V" name="miss.v" ID="13801" frame="
       Success_or_failure" frameID="1388" totalAnnotated="3"
       xsi:schemaLocation="../schema/lexUnit.xsd" xmlns="http://
       framenet.icsi.berkeley.edu" xmlns:xsi="http://www.w3.org/2001/
       XMLSchema-instance">
2      <header>
3          <corpus description="American National Corpus Texts" name=
               "ANC" ID="195">
4              <document description="Berlitz Intro of Dublin" name="
                   IntroOfDublin" ID="23652"/>
5              <document description="Berlitz Where to Go in Hong
                   Kong" name="WhereToHongKong" ID="23771"/>
6          </corpus>
7          <frame>
8              <FE fgColor="FFFFFF" bgColor="FF0000" type="Core"
                   abbrev="age" name="Agent"/>
9              [...]
10             <FE fgColor="FFFFFF" bgColor="00BFFF" type="Core-
                   Unexpressed" abbrev="goa" name="Goal"/>
11             <FE fgColor="FFFFFF" bgColor="A52A2A" type="Peripheral
                   " abbrev="Means" name="Means"/>
12             <FE fgColor="FFFFFF" bgColor="008000" type="Peripheral
                   " abbrev="Place" name="Place"/>
13             [...]
14         </frame>
15     </header>
16     <definition>COD: fail to attend, watch, or participate in.</
           definition>
17     <lexeme POS="V" name="miss"/>
```

Listing 3.1: LU information in FrameNet example sentence annotation

Next, a long description of valence patterns (i.e. the syntactic positions in which individual frame elements can appear) of the lexical unit follows, this information was however not used in our task.

The part of the data that was most important for our task was the sentence annotation which follows after the valence information (see Listing 3.2). First, the text of the sentence is shown (line 3). Then, each sentence is first annotated with Penn Treebank style information which is left out in the example below as we have already seen it in the full text annotation (Listing 2.1, lines 4–17). Next, the frame information is provided (lines 10–16). Note that the name of the frame is not specified – this information can be found in the lexical unit specification (see Listing 3.1). The frame element annotation remains the same as in the full text annotation (lines 13–16). After the end of the first sentence section, next example

sentence follows.

```
1     <subCorpus name="manually-added">
2         <sentence corpID="195" docID="23652" sentNo="1" paragNo="
              12" aPos="0" ID="4100614">
3             <text>Dublin theater is legendary , and no visitor
                  should miss seeing a performance at the Abbey
                  Theatre or Gate Theatre .</text>
4             <annotationSet cDate="11/14/2006 04:25:22 PST Tue"
                  status="UNANN" ID="6539613">
5                 <layer rank="1" name="PENN">
6                     [...]
7                 </layer>
8             </annotationSet>
9             <annotationSet cDate="02/26/2007 11:01:30 PST Mon"
                  status="MANUAL" ID="6541426">
10                <layer rank="1" name="Target">
11                    <label cBy="RLG" end="55" start="52" name="
                          Target"/>
12                </layer>
13                <layer rank="1" name="FE">
14                    <label cBy="RLG" feID="7788" end="43" start="
                          34" name="Agent"/>
15                    <label cBy="RLG" feID="7789" end="113" start="
                          57" name="Goal"/>
16                </layer>
17                [...]
18            </annotationSet>
19        </sentence>
20        <sentence corpID="195" docID="23771" sentNo="1" paragNo="
              137" aPos="0" ID="4106287">
21            [...]
22        </sentence>
23        [...]
24    </subCorpus>
25 </lexUnit>
```

Listing 3.2: Frame information in FrameNet example sentence annotation

3.2 Machine Translation and Information Transfer

After extracting the example sentences from FrameNet, it was necessary to get their Czech translations. This was done using the TectoMT translation system (for further information, see Popel and Žabokrtský [2010]), which performs language transfer on the tectogrammatical layer. During the translation process, the tool creates both a-layer and t-layer tree representation for the English sentence and with the help of these data builds the t-tree and a-tree that represents the same sentence in Czech. Using the trees, it then outputs the Czech translation as a string.[1]

This type of language transfer was very useful for us, as all the trees built during the translation process could be extracted in the treex format (for a description of the format, see Section 2.2 or Žabokrtský [2011]). This allowed us to extract the dependency relations in the Czech sentence. It was also possible to include GIZA++ [Och and Ney, 2003] in the TectoMT tree building process, which added word-level alignment to the trees. Thus, we were able to work with the English and Czech tree pairs in the exact same way we did when transferring the data using PCEDT.

Unfortunately, the TectoMT transfer uses only some of the features that can appear in the trees. One of the features that are almost utterly left out by TectoMT is the analytical function: in the treex files obtained from TectoMT, only subjects and auxiliary tokens were annotated. Therefore, it was not possible to use this feature in our task. Apart from the analytical function, all the features that we considered useful for frame prediction were available in both data resources.

To transfer the frame semantic annotation from FrameNet to the Czech translations, we used a slightly modified version of the script described in Section 2.3. The modifications that had to be made included a change in the way FrameNet annotation is read (this was due to the fact that the data are a little different as described in the previous section). Furthermore, some difficulties arose in the process of matching the annotated phrases from FrameNet to their corresponding phrases in the English part of our MT-created "treebank". While the sentence-level correspondence was perfect (this was due to the fact that the translation input was in a one-sentence-per-line format which was kept by the translation system), it was often difficult to find the correct word correspondence because of a different treatment of hyphens, dashes, slashes, whitespaces etc. A vast majority of these problems could be resolved automatically using regular expressions (see Attachment 1). Only one case, namely the presence of special characters instead of a space, was corrected manually in the data.

[1] The TectoMT translation and tree building has been wholly managed by Mgr. Rudolf Rosa.

3.3 Evaluation

Since the Czech data were entirely automatically created, it was necessary to evaluate them to get a better idea about their quality. However, as the dataset is very large, only a small portion was subject to the evaluation, namely 176 randomly chosen sentences.

Similarly as before (Section 2.4), the evaluation was done manually by just one person. The error types that were distinguished in the first evaluation process also remained the same, with one exception: a new error type was introduced – the error resulting from an incorrect translation. For an illustration of a translation error, see the following example extracted from the sentence "She wouldn't talk to your colleagues, but she'd settle for talking to me." The frame evoking elements are highlighted in boldface and the evoked frame is in brackets right after its evoking element.

(5) *she'd* ***settle****(Make_agreement_on_action) for talking to me.*

(6) *ona o ' d* ***, usaďte se****(Make_agreement_on_action) , že mluvíte*
she o ' d **, sit down**(Make_agreement_on_action) , that talk
se mnou .
with me .

In the example above, the meaning of the frame evoking element has changed because of a wrong translation. Therefore, the Czech phrase that has been annotated as a frame evoking element of the frame Make_agreement_on_action, has a meaning that is quite distant from the frame, the annotation is wrong and the sentence falls into the translation error type.

On the other hand, cases where a different part of the sentence (that is, one that is not and should not be marked as a frame evoking element) is translated wrongly are not marked as any type of errors. Also, if the meaning of the frame evoking element is preserved correctly but the form is wrong, we do not mark the sentence as containing an error. We can afford to be this benevolent because in the current version of FrameNet, only lexical meanings (as opposed to grammatical meanings) evoke a frame.[2] Therefore, it is correct to associate the given frame with any form of the word that evokes it.

[2]In this aspect, the FrameNet differs from the frame semantics theory which states that even the form of a word can evoke a frame. An example of this can be recipes, which have a specific form in many languages. In Czech, the 1^{st} person plural affirmative is the typical verbal form used in recipes (instead of the imperative), which helps a reader distinguish between a recipe and other types of instructions which are more usually written in an imperative form.

The evaluation results can be seen in Table 3.1. The numbers in the second column represent the counts of sentences, the numbers in the third column are the same results expressed in percent. A sentence was considered correct only when every word in the sentence was either correctly annotated as a frame evoking element or correctly annotated as not being one.

	Count	Percent
Correct	126	71
Alignment	34	19
Language differences	5	3
FrameNet errors	1	1
Translation errors	10	6
Multiple error types	0	0
Sum	176	100

Table 3.1: Evaluation results – MT data transfer

As the results show, the machine translated data are comparable with the ones obtained using a human created translation. Therefore, by adding this type of data to our set, we managed to increase its size significantly without seriously damaging their reliability.

4. Automatic Frame Prediction

After we managed to collect a substantial amount of data, we could proceed to our main goal of automatically finding frame evoking elements and predicting the frame they evoke. However, the data needed to be further corrected and filtered first to get rid of at least the easily and automatically detectable errors and the very rare, and therefore unreliable, annotation instances.

4.1 The Data

Although we managed to achieve a fairly high quality of frame annotation transfer, the resulting data still contained many errors. They also comprised a large amount of different frame annotations and while some lemmas were annotated hundreds of times (e. g. *říci* 'to say' had 537 annotated occurrences), other ones had only a very small number of annotated instances in the data (many lemmas with only one annotated occurrence could be found in the data, however, most of those were errors; some of the correct ones were for example *proměnlivý* 'changeable', *míra* 'measure' and a long list of proper names) and their annotation could therefore not be considered very reliable. For both of these reasons, we decided to filter our data to obtain a dataset of reasonable size and quality which could serve as an input for the automatic frame prediction experiments.

First, we sorted all the lemmas that appeared in the data by their number of annotated occurrences, excluding ones that could easily be marked as wrong. For example, in the FrameNet database tokens without a clear and definable meaning such as punctuation marks, pronouns, particles, interjections and a majority of conjunctions and prepositions are not annotated. Therefore, if we found any of the above mentioned parts-of-speech annotated as a frame evoking element, we could immediately discard it.[1] Then, we removed the modal and auxiliary verbs as they often were annotated in the Czech data even though they hadn't been annotated in FrameNet. A most prominent example of this problem is the verb *být* 'to be', which is used as an auxiliary verb forming multiple forms of Czech verbs, e.g. the past tense, as can be seen in the following example from the annotated data:

(7) *I* ***felt****(Perception_experience) the temblor begin*

[1] In the Czech tradition, prepositions and conjunctions in general are usually not considered to have a lexical meaning (see for example [Čermák, 2010, pp. 172–183]), which is why we decided to remove all of their occurrences despite the fact that in FrameNet, some are annotated as frame evoking elements (e.g. the preposition *in* or the conjunction *although*) On the other hand, we decided to keep the annotation of numerals.

(8) ***cítil****(Perception_experience)* ***jsem****(Perception_experience)* *, že přichází*
felt(Perception_experience) $\mathbf{be}_{past}$(Perception_experience) , that comes
otřes
temblor

The form *cítil jsem* is an equivalent of the verb *felt* and is therefore correctly annotated as the frame evoking element. However, the form consists of two words, both of which are analysed and associated with a lemma. Both of these lemmas are then marked as the evoking elements of the frame Perception_experience; this is not convenient for our task because the lemma *být* in its autosemantic occurrence does not evoke this frame. Since the verb *být* is a part of many Czech verbal forms (1^{st} and 2^{nd} person of both singular and plural past tense, all forms of present and past conditional, all forms of future tense of imperfective verbs etc.), this problem was very frequent in the data and the lemma was annotated with 555 frames in the Czech data. Contrary to that, in the FrameNet annotation the verb *to be* is a part of 6 frame evoking elements.

After filtering out the problematic auxiliary and modal verbs, we picked the 100 most frequently annotated lemmas from the remaining ones (the complete list of the lemmas can be found in Attachment 2). Next, we extracted all sentences that contain at least one of these lemmas from the whole data as our dataset for training and testing. When the sentence originated from the PCEDT transfer, we kept it whole as it contains the full text annotation; however, when it came from the MT transfer, we only kept the tokens that were annotated (with the information about their parent and children stored as their features) since most frame evoking elements in the MT-transferred annotation were not annotated (this is due to the example nature of the sentences).[2] All tokens in the chosen sentences which were annotated but did not belong to any of the lemmas we had picked, were marked as unannotated since we were not interested in predicting frames for other lemmas than the 100 most frequent ones. That way, we managed to get a dataset that contained enough occurrences of both the frequently annotated lemmas and the tokens that are not supposed to be annotated.

Then, we divided these filtered "sentences" (this means actual sentences from

[2]The removal of the unannotated tokens from the example sentences causes the resulting data to contain many annotated instances and rather few unannotated tokens compared to what the proportion of the annotated and unannotated tokens would be if the whole data were full-text annotated. However, as we wanted to test the ability of our automatic tools to predict that a lemma should not be associated with a frame, we needed to include at least some instances of lemmas with no annotation. While we realise keeping only the annotated data from the full text annotation makes the resulting dataset rather distant from the way it should normally look, we could not find any better way to deal with the problem. To show how the predictors perform on more balanced data, we present the results of experiments in which all unannotated tokens were left out (marked as "Excluding 'None'" in the result tables or organised in separate tables).

the PCEDT data and annotated parts of the sentences from the MT data) into 10 approximately equal sized sections in order to perform a 10-fold cross validation on our future experiments. Altogether, the data contained 29,401 word tokens, out of which 22,283 were annotated. In total, the dataset comprised 4773 lemma-frame pairs (if we count a lemma and the empty annotation as a pair as well) and 2479 lemmas, i.e. there were on average 1.9 frames per lemma. However, if we only count the annotated instances, we arrive at a dramatically different result: with 2373 annotated frame-lemma pairs and 100 unique annotated lemmas we get the average of 23.7 annotated frames per lemma (for further information about this property of the data, see Chapter 5).

4.2 The Baseline

To set the baseline for the automatic frame prediction task, we created a simple script that finds the most frequent frame for every lemma in the training data and assigns it to every occurrence of the lemma in the test data.

We assumed that this prediction would be fairly successful since the Czech lemmas in the data are already partially disambiguated. The disambiguation is a part of the output of the automatic morphological analyser used in both PCEDT and TectoMT. Each word that is considered ambiguous is marked with a number indicating which of the possible meanings it belongs to, and an explanation of the meaning in brackets after the ^ symbol. The following example shows the disambiguation:

Czech lemma	Meaning
stát-1_^(státní_útvar)	'state (country)'
stát-2_^(něco_se_přihodilo)	'to happen'
stát-3_^(někdo/něco_stojí,_např._na_nohou)	'to stand'

Table 4.1: Example of meaning disambiguation in the Czech data

While the disambiguation certainly helps in the task of frame prediction, we did not expect it to be sufficient to solve the task completely. The disambiguation that appears in the Czech data covers only a part of all the ambiguous words that appear there and is much more coarse-grained than the frame annotation in FrameNet. However, it helped us achieve a fairly high baseline result.

The table below shows the results of the baseline system in both relative counts (number of correctly predicted tokens / number of all tokens) across all the 10 cross-validation test sets and in percentage.[3] In the first row, results that contain

[3]In the "Percent" column (in this table as well as in all other result tables that appear in the

the 'None' annotation are shown – these results include both correctly annotated tokens and those that have correctly been marked as unannotated (not being a frame evoking element). In the second row, the 'None' annotation is excluded and the given numbers represent only the portion of correctly annotated tokens among all those that were marked as frame evoking elements in the data, while the unannotated tokens are left out.

	Relative count	Percent
Including 'None'	20,522/29,401	69.8 ± 2.9
Excluding 'None'	13,441/22,283	60.3 ± 3.1

Table 4.2: Results of the baseline frame prediction tool

The correctness of the predicted frames was checked against the annotation that was a part of the test data (i.e. the annotation resulting from the transfer described in chapters 2 and 3). To get a more accurate assessment of the prediction quality, it would be necessary to get a manually confirmed annotation of the test data which would require further human work.

Given the quality of both the training data and the test data we use for the evaluation, we consider the success rate of the baseline system to be quite high.

4.3 Frame Prediction Using Decision Trees

In attempt to improve the quality of frame prediction, we decided to experiment with machine learning techniques and chose to start with the one that is probably the simplest and most accessible – the decision trees. An important advantage of this method is that, compared to other machine learning methods, it makes it easier for a human to see and interpret the results of the learning process and assess the importance of individual features in the decision making. As far as we know, the decision trees have never been used for frame prediction before. However, they were successfully used in a similar task of valency frame prediction by Semecký [2007] and the prediction of both valency frames and WordNet synsets by Bejček [2006].

thesis), we provide the average percentage of successful tokens in the cross-validation test sets. Furthermore, we state the maximal deviation from this average among the individual 10-fold cross-validation test sets.

4.3.1 Decision Trees

A decision tree is a structure of nodes in the form of a directed rooted tree. It has one root node, some leaf nodes and usually some internal nodes on the path between the root and the leaves. All the nodes except the leaves are called decision nodes. They contain questions (arguments) the answers to which (argument values) correspond to the edges (branches) directed from the given node. The leaves then contain the resulting class, in our case the frame. When classifying an instance of the data (in our case, when assigning a frame to a token), we go through the tree "starting at the top node, testing its question, branching to the appropriate node, and then repeating this process until we reach a leaf node."[Manning and Schütze, 1999, p. 578]

For a simplified illustration of a decision tree, see Figure 4.1 [Bajcsy, 2002] which is a visualization of a decision tree that predicts whether or not to play baseball based on the weather. When deciding, we first need to answer what the outlook is (root node). If it is sunny, we follow the edge labelled "Sunny" and arrive to a node that checks the humidity. After choosing to follow either the edge labelled "High" or the one labelled "Normal", we arrive to the answers "No" and "Yes", respectively. If we follow the "Overcast" edge from the root instead, we immediately reach the "Yes" answer. Following the "Rain" edge will bring us to the node in which a question about the wind is posed. If we follow the edge labelled "Strong", we arrive at a "No", while following the edge labelled "Light" results in a "Yes".

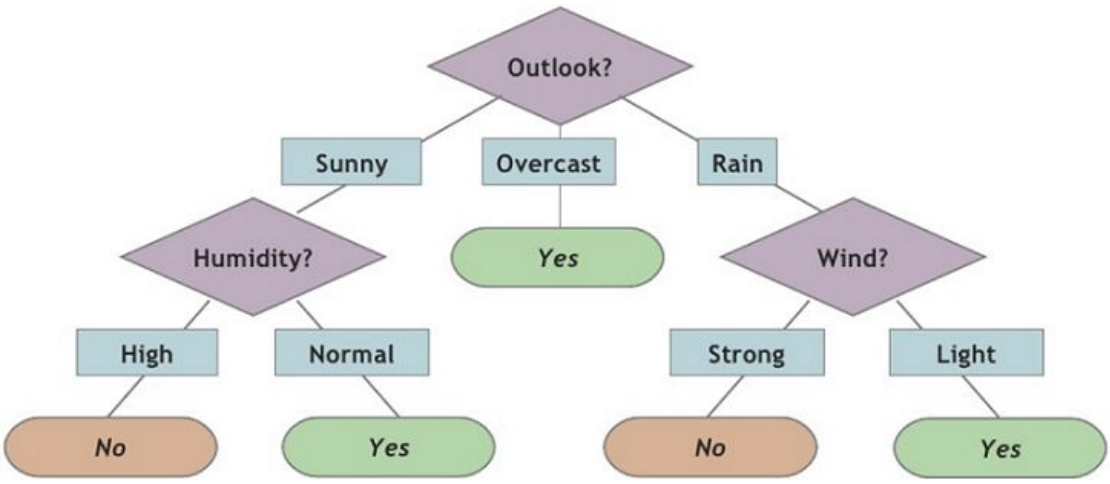

Figure 4.1: A simple decision tree visualisation [Bajcsy, 2002]

There are various training strategies that can be used to create a successful decision tree. Manning and Schütze [1999] state that the most usual strategy is

building a large tree first and then pruning it to a desired size. "The pruning step is necessary because very large trees overfit the training set. Overfitting occurs when classifiers make decisions based on accidental properties of the training set that will lead to errors on the test set (or any new data)."[Manning and Schütze, 1999, p. 582]

In our experiments, we used the AI::DecisionTree module [Williams, 2012] (with only a slight modification that did not influence the classification process[4]), which follows the training strategy described above. In the documentation, the step of creating the initial large tree is described as follows: "The current implementation of this module uses an extremely simple method for creating the decision tree based on the training instances. It uses an Information Gain metric (based on expected reduction in entropy) to select the 'most informative' attribute at each node in the tree. This is essentially the ID3 algorithm, developed by J. R. Quinlan in 1986. The idea is that the attribute with the highest Information Gain will (probably) be the best attribute to split the tree on at each point if we're interested in making small trees."

The ID3 algorithm is based on the notions of entropy and information gain. Entropy characterises a dataset concerning its amount of uncertainty. It is usually labelled H(S) where S is the dataset. It is calculated as follows:

$$H(S) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{4.1}$$

where S is the dataset for which the entropy is being calculated, X is the set of classes in S and p(x) is the probability of the class x in the data S.

The information gain specifies how the entropy of a dataset changes if we add certain information. In our case, the information gain of an attribute A, marked as IG(A), measures the difference in entropy of the dataset S before and after splitting it on the argument A.[5] The information gain is calculated as follows:

$$IG(A) = H(S) - H(S|A) = H(S) - \sum_{y \in Y} p(y)H(y) \tag{4.2}$$

where H(S) is the initial entropy of S, Y is a set of all the subsets created by splitting S on the argument A (these subsets are marked as y), p(y) is the proportion of elements in the subset y to the number of elements in the set S and H(y) is the entropy of the set y.

[4]The library was set to die when a larger portion of the data were not split on a certain node. Since we wanted to allow for such a thing to happen, we changed the 'die' command for a warning written to the error output.

[5]In this context, we use the notion of information gain IG(A) as a different way of expressing mutual information I(S,A).

```
function ID3 (R: the set of attributes,
     C: the set of classes,
     S: a training set) returns a decision tree;

begin
  If S is empty, return a single node with value Failure;
  If S consists of records all with the same value for
     the categorical attribute,
     return a single node with that value;
  If R is empty, return a single node; its value is
     the most frequent of the values of the categorical attribute
     that are found in records of S; [note that then there
     will be errors, that is, records that will be improperly
     classified];
  Let D be the attribute with largest Gain(D,S) among attributes
     in R;
  Let {dj| j=1,2, .., m} be the values of attribute D;
  Let {Sj| j=1,2, .., m} be the subsets of S consisting
     respectively of records with value dj for attribute D;
  Return a tree with root labelled D and arcs labelled
     d1, d2, .., dm going to the respective the trees

  ID3(R-{D}, C, S1), ID3(R-{D}, C, S2), .., ID3(R-{D}, C, Sm);
end ID3;
```

Listing 4.1: ID3 pseudocode (based on Ingargiola [1997])

When building the tree, the ID3 algorithm always selects the argument with the highest information gain among the arguments which have not been used yet. It then splits the data on the selected argument and continues splitting all resulting subsets recursively. The pseudocode above (a slightly modified example taken from Ingargiola [1997]) shows the behaviour of the algorithm.

The next step, pruning, is based on a similar approach as it uses a minimum-description-length criterion. "The Minimum Description Length (MDL) Principle is a relatively recent method for inductive inference that provides a generic solution to the model selection problem. MDL is based on the following insight: any regularity in the data can be used to *compress* the data, i.e. to describe it using fewer symbols than the number of symbols needed to describe the data literally."[Grünwald, 2005, p. 5] Both of these methods lead to the creation of a relatively small tree while keeping the amount of learned information high.

4.3.2 Features

From the data, we were able to extract different kinds of features that could be used for the decision tree training and frame prediction. As we have already

mentioned, probably the most important feature is the lemma. Apart from it, we used the word form, tag (or certain parts of it) and the information about dependency parent and children, namely their tags (or certain parts of them). Even though we would have been able to get further information about the parent and children, such as their lemmas or word forms, we decided to avoid using them because of extreme data sparseness. As the results have shown (see the following subsection), this is a problem that arises with many more features than the ones just mentioned.

4.3.3 Results

To get a thorough idea about which features really are useful for our task, we performed a series of experiments training one tree for the whole data using different sets of features on the same sets of 10 fold cross validation training and test data. The results, sorted according to the accuracy[6] from the least successful feature set to the most successful one, can be seen in Tables 4.4 and 4.5 below.

First, we trained the decision trees using **lemmas** as the only feature to compare the results with our deterministic baseline. Unsurprisingly, the results were very close to the baseline results, i.e. assigning the most frequent semantic frame to the given lemma.

Then, we used **lemma** and each of the remaining features one by one for the training to see which ones help improve the results.

First, we chose the **word form**. We believed that in some cases it could be helpful since in Czech various cases exist that two words have the same lemma but a slightly different paradigm. For example, the word *čistič* 'cleaner' can have two meanings – either a person who cleans something (animate) or a detergent (inanimate). The lemma is the same in both cases but the words differ in some forms, e.g. the singular accusative, which is *čističe* in the animate paradigm and *čistič* in the inanimate one. Since one of the words denotes a person and the other one denotes a substance, they are very likely to evoke different frames. However, only a small portion of words in Czech display this property and even if they do, only a few forms in the whole paradigm differ in the way described above. This is why adding word forms to the features did not bring any improvement. On the contrary, probably due to the data sparseness, it made the results worse.

Next, we chose the **part of speech (POS)** of the word. In some cases, this could also improve the results since some words in Czech can have the same lemma although they belong to a different part of speech. An example of this can be the word *podle*, which can either be a preposition meaning 'according to' or an adverb

[6]By *accuracy* we mean the number of correctly predicted tokens divided by the number of all predicted tokes.

meaning 'meanly'. However, some of these sets of words with an identical lemma, especially those in which the individual words have a very different meaning (like the word *podle* mentioned above), are already disambiguated in the lemma feature as seen in Section 4.2. Also, similarly as in the case of the word forms, words that would share a lemma and could be distinguished using the part of speech are quite rare in Czech. This is why the addition of POS brought only a slight improvement. However, since the data sparseness problem was avoided (the POS position of the morphological tag can have only 12 different values), the inclusion of the POS did not damage the results.

Furthermore, we tried to add the **POS of the parent node** among the features. While we could not think of any clear and obvious way in which this could help, we assumed that the nodes that are close (in the dependency tree) to the one we are observing could give us some useful additional information. Since it was not possible to use the lemma of the parent due to the data sparseness, we tried the POS instead. Interestingly, the results were better than with the lemma alone, even though only slightly.

Lastly, we used the **morphological cases of the child nodes** as additional features. We believed that these could be of some help as the cases of the dependency children are a realisation of valency patterns which are closely related to word senses (for a closer look at this phenomenon in verbal valency, see for example the works of Hajič et al. [2015], Semecký [2007], Semecký and Podveský [2006] or Bejček [2006]). Since Czech is a free-word-order language, we could not rely on the ordering of the child nodes, and due to the fact that many obligatory valency elements can be elided in the surface structure of a Czech sentence (typically the subject) while other, non-obligatory elements can be present, we could not use their counts either. Therefore we decided to create 9 features (one for every value the case position of the morphological tag can have):

Value	Description
-	Not applicable
1	Nominative
2	Genitive
3	Dative
4	Accusative
5	Vocative
6	Locative
7	Instrumental
X	Any

Table 4.3: Possible values of the case position in the morphological tag.

The values of these features were set to 0 by default and if a child of a certain case was encountered, the corresponding feature value was changed to 1. The addition of these features led to an improvement in the results, as can be seen in tables 4.4 and 4.5 in the row named "Lemma + all children".

However, we believed the results could be further improved by leaving out the cases that can cause data sparseness problems or do not have a great value for our task. This is why we excluded the "any" and "not applicable" cases. We also left out the nominative (1^{st} case), which is typically the case of the subject. As we have mentioned before, the subject in Czech sentences tends to be left out quite often. Besides, a vast majority of Czech verbs require a subject, which is why its presence in the valency frame does not bring much new information. Furthermore, we excluded the vocative (5^{th} case) which is a case used for addressing a conversation partner or other addressee of the utterance. It is usually not considered a part of the sentence structure, and therefore also does not belong to the valency frames of other words. After excluding these cases, the frame prediction quality moved further high, reaching 62.1% in the results that disregard the unannotated words (see Table 4.5, row "Lemma + children 23467").

After finding out which features can help improve the results, we further experimented with their combinations. Of all the experiments, we managed to surpass the results of the "Lemma + children 23467" only once, namely with the combination "Lemma + children 23467 + POS". However, we have achieved only a slight improvement.

Used features	Relative count	Percent
Baseline	20,522/29,401	69.8 ± 2.9
Lemma + form	20,312/29,401	69.1 ± 2.6
Lemma	20,539/29,401	69.9 ± 2.9
Lemma + POS	20,543/29,401	69.9 ± 3.0
Lemma + parent POS	20,575/29,401	70.0 ± 2.5
Lemma + all children	20,800/29,401	70.7 ± 2.7
Lemma + children 23467	20,926/29,401	71.2 ± 2.5
Lemma + ch. 23467 + POS	**20,943/29,401**	**71.2 ± 2.4**

Table 4.4: Results of the frame prediction using one decision tree – including 'None'

In an attempt to further improve the results, we decided to train a separate tree for each lemma (as was done by Semecký [2007]). This was motivated by the idea that when only one tree is trained, it is possible for a training instance with a certain lemma to be assigned to a class (in our case, frame), which is not

Used features	Relative count	Percent
Baseline	13,441/22,283	60.3 ± 3.1
Lemma + form	13,257/22,283	59.5 ± 2.5
Lemma	13,458/22,283	60.4 ± 3.1
Lemma + POS	13,462/22,283	60.4 ± 3.2
Lemma + parent POS	13,494/22,283	60.6 ± 2.5
Lemma + all children	13,719/22,283	61.6 ± 2.5
Lemma + children 23467	13,845/22,283	62.1 ± 2.6
Lemma + ch. 23467 + POS	**13,862/22,283**	**62.2 ± 2.5**

Table 4.5: Results of the frame prediction using one decision tree – excluding 'None'

associated with the lemma in the data. This could happen if the decision tree used other features than the lemma in the early stages of the decision.

However, training the trees separately for each lemma did not bring any important improvement in the results of the majority of our experiments. An observation of the structure of the large tree trained for all lemmas explained why: the lemma was usually picked as the attribute with the highest information gain, and its value was therefore checked in the root node. After splitting on the lemma, the tree was divided into subtrees which approximately corresponded to the trees that resulted from training for each lemma separately. The only case where this was not true was the experiment in which we used the combination of lemma and form as features (marked Lemma + form in tables 4.4 and 4.5 and Form in tables 4.6 and 4.7), since the large decision tree chose the form as the first splitting argument. In this case, the training of separate trees yielded an improvement in the data. However, as this was still one of the the least successful feature combinations, the improvement is of little interest.

Used features	Relative count	Percent
Lemma	20,524/29,401	69.8 ± 2.9
Form	20,463/29,401	69.6 ± 2.3
POS	20,544/29,401	69.9 ± 3.0
Parent POS	20,600/29,401	70.1 ± 2.5
All children	20,803/29,401	70.8 ± 2.8
Children 23467	20,803/29,401	70.8 ± 2.5
Children 23467 + POS	**20,935/29,401**	**71.2 ± 2.5**

Table 4.6: Results of the frame prediction using separate decision trees – including 'None'

Compared to the results of Bejček [2006] and Semecký [2007], our results were rather disappointing. Bejček [2006] reaches the accuracy of 88.803% in the task

Used features	Relative count	Percent
Lemma	13,443/22,283	60.3 ± 3.0
Form	13,379/22,283	60.0 ± 2.3
POS	13,463/22,283	60.4 ± 3.2
Parent POS	13,519/22,283	60.7 ± 2.5
All children	13,722/22,283	61.6 ± 2.6
Children 23467	13,851/22,283	62.2 ± 2.6
Children 23467 + POS	**13,854/22,283**	**62.2 ± 2.6**

Table 4.7: Results of the frame prediction using separate decision trees – excluding 'None'

of synset prediction for nouns, 93.082% for adjectives and 86.5% in the task of valency frame prediction for verbs. However, it has to be noted that Bejček's baseline which, like the one we use, assigns the most frequent synset/frame to each lemma, yields results with the accuracy of 87.362%, 92.060% and 84.87% for nouns, adjectives and verbs respectively. Thus, the difference between the baseline and the decision trees is not so high. The reason why both the baseline and the decision tree results are much better than ours is that the data in Bejček's experiments had been annotated manually and the instances where all annotators reached an agreement were picked. Thus, the quality of the data is much higher than the quality of the ones we use (for a closer look at the problems we encountered and the way a simple manual correction of the annotation improved our results, see Chapter 5).

Semecký [2007] reached the accuracy of 77.48% using decision trees with the ID3 algorithm against the baseline of 68.27%. The reason why the baseline is higher than ours[7] is again related to the quality of the data: while we used automatically created features and dependencies and mostly also machine translated sentences, Semecký [2007] uses manually created data. The reason why Semecký [2007] managed to reach such an improvement with the decision trees, is probably related to his choice of features. His most successful experiment with ID3 decision trees uses syntactic features such as specific dependent prepositions and conjunctions, but also idiomatic features.[8] In total, he presents hundreds of such features. While this approach proved useful when predicting valency frames of verbs, we assume it would be necessary to largely expand the list of lemmas whose

[7]Note that in the task of valency frame prediction, no "None" or empty class is possible. Therefore, Semecký's results can only be compared with the results we reached when excluding empty frames.

[8]Most of the features, both syntactic and idiomatic, are boolean and capture whether a certain lemma is among the dependencies of a verb or not.

presence among the dependencies is marked by the features, in order to achieve better results for nouns, adjectives and adverbs as well.

4.4 Frame Prediction Using Support Vector Machines

In addition to the decision tree experiments, we decided to use support vector machines (SVM) for the frame prediction. Our main motivation was to get a comparison of the suitability of both machine learning methods for the task of frame prediction. While decision trees seemed like a good choice because of their understandability, the advantage of the SVM lies in the fact that they have already been used in similar experiments in various languages and yielded interesting results. For example, Pradhan et al. [2004] used the SVM for shallow semantic parsing of English sentences but unlike us, they used the PropBank style annotation and annotated the whole argument structure, while we only focus on the identification of frame evoking elements. A similar approach has been used for the FrameNet-style predicate structure annotation of Swedish by Johansson and Nugues [2006] who later also implemented a SVM-based frame semantic analyser for English [Johansson and Nugues, 2007].

The latter tool shares many properties with the ones we present here. Most importantly, a part of the analysis performed by the system is identical with our task – the prediction of evoked frames. Furthermore, the analyser extracts many features from dependency trees, such as form and lemma of the target word, a list of its children and the parent word. It also filters out problematic lemmas in a similar manner as we have, even though it allows some prepositions and auxiliary verbs to be annotated if they fulfil certain conditions while we automatically exclude them all.

This tool reached a 84% accuracy in target word disambiguation against the 74% score of the baseline which, like the one we chose, assigned the most frequent frame to each lemma.

4.4.1 Support Vector Machines

Support vector machines (SVM) are based on the assumption that the data we are trying to classify can be represented as points in an n-dimensional space. For simplicity, let us focus on points in a plane and assume our classification task is restricted to a two-class problem. Our goal is to find a hyperplane (in our case a line) which can separate the data in one class from the data in the other class. If this can be done perfectly, i.e. all instances belonging to one class are on one side of the line and all instances of the other class are on the other side, the data are

called *linearly separable*. This is the ideal case for a SVM. However, even if the data are not linearly separable, we can still perform the classification using the SVM if we allow for a small portion of the data to be on the "wrong" side of the separating line.

If the data are linearly separable, there can be multiple ways to draw the line between them (see Figure 4.2 which was taken from Harrington [2012, p. 103]). Among these possibilities, we want to choose the one which splits the data in the most robust way. To do this, we have to find the largest possible *margin*, i.e. the largest possible distance between the separating hyperplane and the data point that is closest to it. The reason why the largest possible margin is most convenient is that the classifier works in such a way that "the farther a data point is from the decision boundary, the more confident we are about the prediction we've made."[Harrington, 2012, p. 103] Thus, in Figure 4.2 below, the best separating hyperplane is the one in graph D.

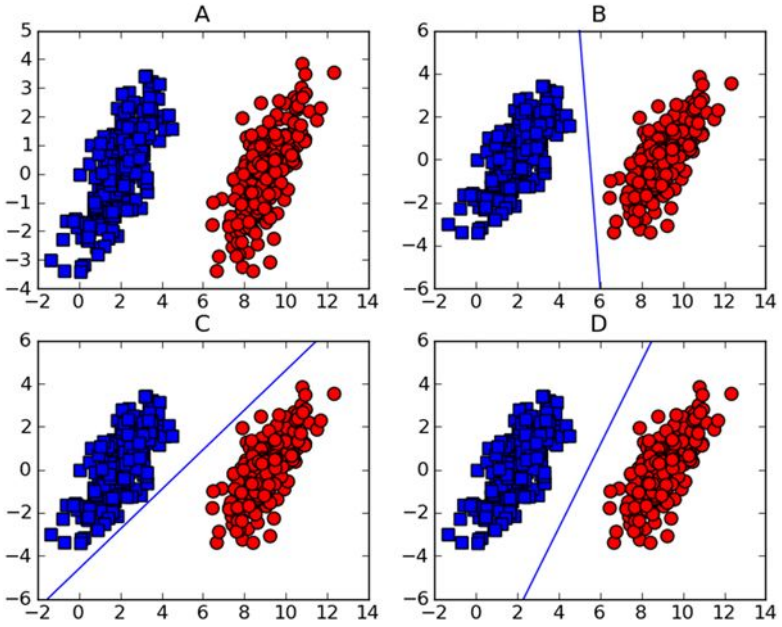

Figure 4.2: Linearly separable data and different separating hyperplanes [Harrington, 2012, p. 103]

The points that are closest to the separating hyperplane are called *support vectors*. The goal of the SVM is to maximise the distance between the support vectors and the separating hyperplane.[9]

[9]For a closer description of the methods that can be used for the SVM training, see Harrington [2012, pp. 101–180]

For this task, we used a popular SVM library, LIBSVM [Chang and Lin, 2011], namely its Perl-adjusted version by Spencer et al. [2012] with default settings. Since the SVM require numeric values of all features, we mapped the lemmas and parts-of-speech on positive integers in an arbitrary way using their order of first appearance in the data.

4.4.2 Results

To train and test our SVM predictor, we used the same data as in the baseline and decision tree experiments, divided into the same 10-fold cross-validation sets.

The results of the experiments are very similar to the ones we achieved using the decision trees, as can be seen in tables 4.8 and 4.9 below. Note that we leave out the results of SVM training using the lemma as the only feature. The reason for this is that such a SVM would be required to split data points on a line and since one frame (one class) usually corresponds to multiple different lemmas in the data, the points are not separable, i.e. instances of one class can appear on many different places along the line. An experimental training of a SVM that only used the lemma feature resulted in the majority of lemmas being labelled with the same frame yielding an accuracy lower than 0,1%.

As the results show, the SVM and the decision trees performed comparably well on our data. Interestingly, the combination "Lemma + all children" yielded more accurate results than "Lemma + children 23467" when the unannotated instances were excluded. However, as the difference in accuracy is very low, we believe that this unexpected result is accidental and does not point to any special properties of the SVM prediction.

In general, the SVM performed slightly better than the decision trees. However, the results of both machine learning methods are so close that it would not be safe to conclude that the SVM are more suitable for the task of frame prediction.

When comparing our results with the ones achieved by Semecký [2007], we found out our SVM predictor performed rather poorly. However, we also noticed that the difference between his results with ID3 decision trees and with SVM is similar to ours – Semecký [2007] reports a 77.48% accuracy with the decision trees and a 78.19% accuracy when using SVM.[10] Thus, we assume that similarly as in the case of decision trees (see Subsection 4.3.3), the difference between Semecký's results and ours is caused by the incomparable quality of the data and the feature selection.

[10]Both of these results were achieved using the same feature sets and are best among the results of the given ML method.

Used features	Relative count	Percent
Baseline	20,522/29,401	69.8 ± 2.9
Lemma + form	20,392/29,401	69.4 ± 2.6
Lemma + POS	20,530/29,401	69.8 ± 2.9
Lemma + parent POS	20,527/29,401	69.8 ± 2.9
Lemma + all children	20,849/29,401	70.9 ± 2.6
Lemma + children 23467	20,888/29,401	71.0 ± 2.5
Lemma + all children + POS	**20,978/29,401**	**71.4 ± 2.3**

Table 4.8: Results of the frame prediction using SVM – including 'None'

Used features	Relative count	Percent
Baseline	13,441/22,283	60.3 ± 3.1
Lemma + form	13,307/22,283	59.7 ± 2.7
Lemma + POS	13,449/22,283	60.4 ± 3.1
Lemma + parent POS	13,447/22,283	60.3 ± 3.1
Lemma + children 23467	13,900/22,283	62.4 ± 2.6
Lemma + all children	13,935/22,283	62.5 ± 2.4
Lemma + all children + POS	**13,949/22,283**	**62.6 ± 2.4**

Table 4.9: Results of the frame prediction using SVM – excluding 'None'

5. Back to the Data

5.1 Random Frame Selection

In order to explain why using machine learning methods and adding various kinds of features brought only a small improvement over the baseline, we tried to assess how well the baseline performs with respect to the data. To do this, we conducted another experiment in which the predictor chose the predicted frame for every lemma in the test data randomly from all the frames that were associated with the given lemma in the training data. Each lemma-frame pair from the training data was considered at most once by the predictor, regardless of how many times it had been observed in the training data. Thus, the set of frames associated with a certain lemma had a uniform distribution, i.e. all of the frames that appeared with a given lemma were equally likely to be predicted.

With these settings, our predictor performed poorly reaching a 30.8% accuracy on all tokens and only a 8.9% accuracy when the unannotated ones were disregarded. This shows that the baseline, which assigns the most frequent frame to each lemma, is already based on a type of information that says a lot about the data. Checking the lemma-frame pair frequency in the data explained why choosing the predicted frame randomly from all possible frames for a lemma is a bad strategy: most of the lemmas are associated with quite a large number of frames (23.7 on average), most of which only appear once or twice with the given lemma. Only very few lemma-frame pairs have a reasonable amount of occurrences in the data.[1]

The graph in Figure 5.1 illustrates this phenomenon. The x axis represents the number of lemma-frame pair occurrences in the data and the y axis indicates the number of lemma-frame pairs with the given number of occurrences (in this section, we use the term *frequency* to refer to the latter type of information). Due to the fact that the frequency falls dramatically with a rising value on the x axis, we decided to use a $\log_{10}$ scale on both axes. As the figure shows, the number of lemma-frame pairs with only one occurrence is very high (namely 1,488).

Note that if such a pair appears in the test data, it always results in an error with both the baseline and the decision tree predictors, as the frame has not been observed for the given lemma in the training data and thus can not be predicted. Since we use each of the data instances as a part of the test set at some point (this

[1]The poor quality of the data was quite surprising for us considering our previous manual evaluation of a randomly chosen sample from the MT data and one of the six articles from the PCEDT data. However, since the checked MT data sample was rather small, the quality it has probably does not reflect the quality of the whole data very well. As the MT data comprise the majority of the dataset, their quality is crucial for the quality of the whole set.

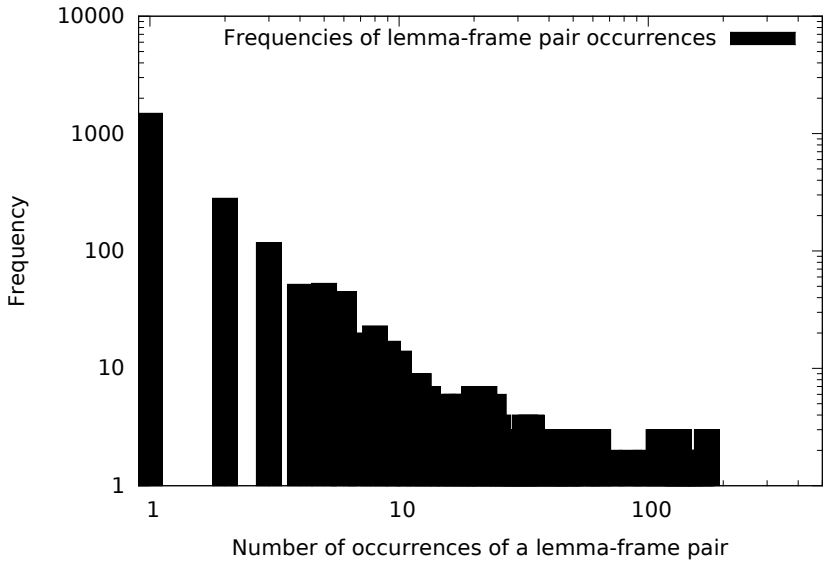

Figure 5.1: Distribution of the numbers of lemma-frame pair occurrences

is the principle of a cross validation), we automatically get 1,488 errors resulting from the one-instance issue. Since our data contain 22,283 annotated tokens, this type of automatic errors forms 6.7% of the results disregarding the unannotated tokens.

5.2 Results on Improved Data

In order to deal with the problem described above, we attempted to filter out some of the problematic instances from the data. We decided to do this manually, since the number of occurrences was not always enough to indicate whether the given lemma-frame pair was correct. This can be illustrated using two examples.

First, let us look at a lemma-frame pair with quite a high number of occurrences all of which were wrong. In our data, the word *řada* meaning 'line' or 'queue' was annotated with the frame Quarreling 22 times, even though it seemed rather unlikely that a word denoting a line would evoke this frame. The reason was that the word *řada* is one of the possible translations of the English word *row* which, in one of its meanings, can also denote a quarrel. Thus, a regularly appearing translation error created a prominent lemma-frame pair which was however completely wrong. Similarly, the word *případ* 'case', 'cause' or 'instance' was marked as an evoking element of the frame Containers 32 times (resulting from a wrong translation of the noun *case*), the adjective *velký* 'big' was associated with the frame Detaining due to a piecewise translation of the idiomatic expression *at large* (21 times) and

the adjective *vysoký* 'high', 'tall' was marked as an evoking element of the frame Intoxication 12 times.

On the other hand, there were also cases when a lemma-frame pair only appeared once but the annotation was correct. For example, the word *plný* 'full' was only once annotated as an evoking element of the frame Biological_urge as the FrameNet lexical unit with this particular meaning had one example phrase, namely "Too full for dessert." Since the Czech word used for describing this amount of saturation indeed is *plný*, we consider this annotation correct and kept it in the data, even though it meant we would get an automatic error in the machine learning experiments.

However, we did not always look at the whole sentences to assess whether the frame-lemma pair is correct – we could not afford to do so since the data are quite large. Instead, we went through all the lemma-frame pairs and discarded those where we could not imagine any circumstances under which they could be correct (1,904 lemma-frame pairs in total). This means that we probably filtered out some of the correct instances believing them to be wrong and kept some of the wrong ones but we consider this a good first step towards the improvement of the quality of our data.

Used features	Relative count	Percent
Baseline (before)	20,522/29,401	69.8 ± 2.9
Baseline (after)	20,525/27,497	74.6 ± 3.3
DT: Lemma + children 23467 + POS (b)	20,943/29,401	71.2 ± 2.4
DT: Lemma + children 23467 + POS (a)	20,936/27,497	76.1 ± 2.7
SVM: Lemma + all children + POS (b)	20,849/29,401	70.9 ± 2.3
SVM: Lemma + all children + POS (a)	**20,932/27,497**	**76.1 ± 2.7**

Table 5.1: Results of the prediction on the data before (b) and after (a) the manual correction – including 'None'.

Used features	Relative count	Percent
Baseline (before)	13,441/22,283	60.3 ± 3.1
Baseline (after)	13,444/20,379	66.0 ± 3.7
DT: Lemma + children 23467 + POS (b)	13,862/22,283	62.2 ± 2.5
DT: Lemma + children 23467 + POS (a)	13,855/20,379	68.0 ± 3.0
SVM: Lemma + all children + POS (b)	13,935/22,283	62.5 ± 2.4
SVM: Lemma + all children + POS (a)	**13,910/20,379**	**68.3 ± 3.0**

Table 5.2: Results of the prediction on the data before (b) and after (a) the manual correction – excluding 'None'.

Using the partially manually corrected data resulted in a notable improvement in accuracy for all automatic prediction methods. Tables 5.1 and 5.2 above show the results of the training using the most successful feature combinations for each prediction method. It also contains the results on the initial data (marked "before" or "b") for comparison.

Conclusion

In the thesis, we tested the possibilities of automatic full-text frame prediction for Czech. Since no data containing Czech full text FrameNet annotation were available, we needed to transfer the information from the English FrameNet [FrameNet Data] first. This was done in two ways.

First, we identified some of the Wall Street Journal Texts from the PropBank Project [Palmer et al., 2005] both in the FrameNet database and in the Prague Czech-English Dependency Treebank (PCEDT) [Hajič et al., 2012], which is a parallel treebank of English texts and their manual Czech translations. Then we copied the FrameNet annotation to the English part of the PCEDT and using the word-to-word alignment, we then transferred the annotation to the Czech part, obtaining a quite reliable but rather small set of data.

Second, we extracted all example sentences from the FrameNet database and translated them to Czech with the TectoMT machine translation tool [Popel and Žabokrtský, 2010]. Using the Giza++ [Och and Ney, 2003], we aligned the corresponding English and Czech sentences word-to-word, which enabled us to transfer the FrameNet information in the same way as when using the parallel treebank.[2] The resulting data were less reliable than the manually translated ones but their quality still appeared to be sufficient for our task. On the other hand, the machine-translated dataset was notably larger which was its main advantage.

After having gathered the data we filtered it in order to get rid of some easily detectable errors as well as rare, and therefore unreliable, annotated lemmas (we chose a dataset of 100 most frequent lemmas, all of which had at least 160 annotated instances in the data). Then, we created a baseline prediction tool which chose the most frequent frame for each lemma in the training data and assigned this frame to every instance of the given lemma in the test data. The baseline reached a 60.3% accuracy on the annotated lemmas.

Subsequently, we proceeded to machine learning experiments. We chose the decision trees to begin with, mainly because their structure can be easily observed and interpreted by human readers. Using different sets of features as well as different ways of training the tree (or separate trees), we reached a 62.2% accuracy on the annotated lemmas as our best result.

The second machine learning method we used were the support vector machines (SVM). After repeating the series of experiments with the same data and feature sets which we used to train the decision trees, we arrived at similar results reaching a 62.6% accuracy.

[2]The TectoMT translations and the Giza++ alignment have been provided by Mgr. Rudolf Rosa.

To explain why the machine learning yielded only slightly better results than the simple baseline, we inspected our data a little more and discovered that a very large portion of the data is comprised of lemma-frame pairs with very few occurrences. Usually, one lemma was associated with a small number of frames which occurred frequently in the data, but also with a large number of frames with only 1 or 2 instances in the data. This is why the baseline yielded highly accurate results – it focused on the frequent, and therefore reliable, frames while ignoring the less frequent ones which were also less reliable (and usually wrong). Following the data inspection, we performed a simple manual correction of the annotation in which we discarded those lemma-frame pairs which we considered wrong under any circumstances. Thus, we managed to reduce our data by 1,904 wrongly annotated instances and achieved an improvement in automatic frame prediction. While the baseline moved up to 66.0%, the decision trees reached 68.0% and the SVM even 68.3%.

Considering the positive impact the simple correction had on the results of all automatic prediction methods, we believe that a further and more thorough manual correction could help reach an even higher accuracy.

Interestingly, the manual data correction did not help improve the performance of the machine learning methods compared to the baseline. Therefore, it seems that the features we chose were not sufficient to increase the accuracy of the machine learning. However, comparing our feature set with the one used by Johansson and Nugues [2007], who reached an 84% accuracy using SVM prediction against a 74% baseline score, we can see only one difference: Johansson and Nugues [2007] used the dependency labels (which correspond to the analytical function in the PCEDT), while we did not because the machine translation output did not contain them. Thus, it appears that the addition of the analytical function labels (manually or using an automatic parser) could be of help when improving the machine learning performance. A similar effect could be reached by including information about valency of the lemmas as a feature (possibly exploiting the VALLEX database [Lopatková et al., 2008]), even though this would probably only change our problem to the problem of predicting the correct valency frame. Also, the VALLEX database currently only contains information about the valency of verbs and some nouns, and therefore can not help us in the task of predicting frames for adjectives, adverbs etc. It would also be possible to use the information about core frame elements in the annotated sentences. In a way, the core elements are related to obligatory arguments in a valency frame – both of these role types capture the fact that a verb requires certain words to appear in the deep structure of a sentence in which it is used. Therefore, the information about core elements represents a very similar property of the verb as the valency frame. Since the information about frame elements is included in the annotation we worked with

in our experiments, this would probably be the easiest way to get this type of information.

The reason why adding the features was not as successful as we had hoped can also possibly lie in the fact that some of them can be wrong. Since all of the features (lemmas, cases, parts of speech and the dependency tree structure) were created automatically, a certain portion of errors has to be anticipated. However, only a thorough manual evaluation and correction of the data could show how many errors there actually are and experiments with the improved data could inform us how much the eventual errors influenced the quality of our prediction.

Furthermore, comparing our results with the ones by Semecký [2007], who reached a 77.48% accuracy with decision trees (using the same algorithm as we did) and a 78.19% accuracy with SVM against a 68.27% baseline in the task of valency frame prediction for verbs, we can see that a different approach to the feature sets could also yield interesting results: in his best performing DT and SVM configuration, Semecký [2007] used a list of important prepositions and conjunctions which can appear as the children of a certain node, as well as a list of idiomatic expressions whose presence among the children of a verb also brings important information about its meaning. However, since Semecký [2007] only worked with verbs while we did not focus on any particular part of speech,[3] it would be necessary to thoroughly investigate the combinatorial and idiomatic properties of nouns, adjectives and adverbs for us to be able to carry out experiments similar to his on our data.

Since we did not manage to apply these improvements in our present experiments, we leave them as suggestions for future work.

[3]We discarded pronouns, prepositions, conjunctions, particles, interjections and punctuation, since these parts of speech usually do not carry lexical meaning.

Bibliography

About FrameNet. https://framenet.icsi.berkeley.edu/fndrupal/about [Accessed 05–26–2016].

Petr Bajcsy. Introduction to Data Mining. University Lecture, 2002.

Eduard Bejček. Automatické přiřazování významu - "sense-tagging". Master's thesis, ÚFAL MFF UK, Praha, 2006.

František Čermák. *Lexikon a sémantika*. NLN, Prague, Czech Republic, 2010.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Charles J. Fillmore. The Case for Case. In E. Bach and R. Harms, editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart, and Winston, 1968. URL http://www.icsi.berkeley.edu/pubs/ai/casefor68.pdf.

Charles J. Fillmore. Frame Semantics and the Nature of Language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech, Vol. 280*, pages 20–32, 1976. URL http://www.icsi.berkeley.edu/pubs/ai/framesemantics76.pdf.

Charles J. Fillmore. Frame Semantics. In *Linguistics in the Morning Calm*, pages 111–137, Seoul, South Korea, 1982. Hanshin Publishing Co.

Charles J. Fillmore and Beryl T. S. Atkins. Towards a Frame-Based Organization of the Lexicon: The Semantics of RISK and Its Neighbors. In A. Lehrer and E. Kittay, editors, *In Frames, Fields, and Contrast: New Essays in Semantics and Lexical Organization*, pages 75–102. Lawrence Erlbaum Associates, 1992. URL http://www.icsi.berkeley.edu/pubs/ai/towarda92.pdf.

FrameNet Data. https://framenet.icsi.berkeley.edu/fndrupal/framenet_data [Accessed 05–26–2016].

FrameNet Index of Lexical Units. https://framenet.icsi.berkeley.edu/fndrupal/luIndex [Accessed 05–26–2016].

Full Text Index. https://framenet.icsi.berkeley.edu/fndrupal/fulltextIndex [Accessed 05–26–2016].

Peter Grünwald. A Tutorial Introduction to the Minimum Description Length Principle. In *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2005.

Jan Hajič. *Disambiguation of Rich Inflection (Computational Morphology of Czech)*. Karolinum, Charles Univeristy Press, Prague, Czech Republic, 2004.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. Prague Czech-English Dependency Treebank 2.0, 2012. URL `http://hdl.handle.net/11858/00-097C-0000-0015-8DAF-4`. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

Jan Hajič, Ondřej Dušek, Eva Fučíková, Zdeňka Urešová, Martin Popel, and Jana Šindlerová. Using Parallel Texts and Lexicons for Verbal Word Sense Disambiguation. *Depling 2015*, page 82, 2015.

Peter Harrington. *Machine Learning in Action*. Manning Publications Co., Greenwich, CT, USA, 2012. ISBN 1617290181, 9781617290183.

Giorgio Ingargiola. The ID3 Algorithm. University Lecture, 1997.

Introduction to PCEDT. `https://ufal.mff.cuni.cz/pcedt2.0/` [Accessed 05–19–2016].

Richard Johansson and Pierre Nugues. A FrameNet-based Semantic Role Labeler for Swedish. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 436–443. Association for Computational Linguistics, 2006.

Richard Johansson and Pierre Nugues. LTH: Semantic Structure Extraction Using Nonprojective Dependency Trees. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 227–230, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1621474.1621522`.

Markéta Lopatková, Zdeněk Žabokrtský, and Václava Kettnerová. *Valenční slovník českých sloves*. Univerzita Karlova v Praze, Nakladatelství Karolinum, Praha, 2008. In cooperation with Karolína Skwarska, Eduard Bejček, Klára Hrstková, Michaela Nová and Miroslav Tichý.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-13360-1.

Mitchell P. Marcus, Beatrice Santorini, and Mary A. Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *COMPUTATIONAL LINGUISTICS*, 19(2):313–330, 1993.

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

Martha Palmer, Dan Gildea, and Paul Kingsbury. The Proposition Bank: A Corpus Annotated with Semantic Roles. *Computational Linguistics Journal*, 31 (1), 2005.

Martin Popel and Zdeněk Žabokrtský. Tectomt: Modular NLP Framework. In *Proceedings of IceTAL, 7th International Conference on Natural Language Processing*, pages 293–304, Reykjavík, Iceland, 2010.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. Shallow Semantic Parsing Using Support Vector Machines. In *Proceedings of NAACL-HLT 2004*, 2004.

Jiří Semecký. *Verb Valency Frames Disambiguation.* PhD thesis, Univerzita Karlova v Praze, Prague, Czech Republic, 2007.

Jiří Semecký and Petr Podveský. Extensive study on automatic verb sense disambiguation in czech. In Petr Sojka, Ivan Kopeček, and Karel Pala, editors, *Proceedings of Text, Speech and Dialogue. 9th International Conference, TSD*, volume 4188 of *Lecture Notes in Computer Science*, pages 237–244, Berlin / Heidelberg, 2006. Springer.

Petr Sgall. *Generativní popis jazyka a česká deklinace.* Academia, Prague, Czech Republic, 1967.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects.* Reidel, Dordrecht, 1986.

Cory Spencer, Matthew Laird, and Saul Rosa. Algorithm::SVM version 3.12. Github, 2012.

Ken Williams. AI-DecisionTree-0.11. CPAN, 2012.

Zdeněk Žabokrtský. Treex – an Open-source Framework for Natural Language Processing. In *Information Technologies – Applications and Theory*, pages 7–14, 2011.

List of Figures

List of Tables

Attachments

Attachment 1 – Resolved Data Errors

PCEDT/TectoMT	FrameNet
word1-word2	word1 - word2
word1 – word2	word1–word2
word1 – word2	word1-word2
word1/word2	word1 / word2
word1 : word2	word1:word2
word1 ; word2	word1;word2
’word	’ word
‘word	‘ word
word’	word ’
word‘	word ‘
word ’ s	word’s
. word	.word
, word	,word
word .	word.
word . .	word..
word . . .	word…
word !	word!
word ?	word?
got ta	gotta
”	“
’’	“
’	‘
-	–
–	-
--	–
. . .	…

Attachment 2 – List of 100 Most Frequently Annotated Lemmas

Counts	Lemmas	Counts	Lemmas	Counts	Lemmas
422	zbraň	218	ostrov	186	výroba
386	útok	217	program	184	nikdy
376	velký	217	uvést	183	sledovat
349	oko	213	raketa	182	hodit
335	cesta	212	plavat	180	růst
333	dítě	211	muset	178	bomba
331	silný	209	část	176	řada
305	dobrý	208	cíl	175	kniha
291	zařízení	205	kritizovat	175	zmizet
286	rok	205	spravedlivý	174	vliv
281	odsoudit	203	jméno	174	zabít
281	příjemný	201	říkat	174	získat
276	pohled	200	mnoho	173	snadný
270	snažit	199	jaderný	172	jistý
269	vědět	199	hlavní	172	tvář
265	vysoký	199	zajímavý	172	zasáhnout
265	začít	199	odpověď	171	malý
258	vidět	198	další	171	vyhnout
256	dohoda	198	skvělý	169	šat
256	žít	198	starý	167	odpovědět
245	dům	197	dva	167	touha
245	křičet	197	město	167	učitel
245	práce	197	úspěšný	166	rodina
244	život	196	síla	166	vytvořit
242	cítit	195	dostatečný	165	slyšet
241	země	192	plný	164	stále
240	vést	191	historie	162	odpor
232	zjistit	191	místo	161	prohlášení
226	jeden	191	špatný	161	použít
224	zpráva	190	vyjádřit	160	úsměv
222	případ	189	opustit		
221	vzít	189	ruka		