Verb Valency Frames Disambiguation

Jiří Semecký

Institute of Formal and Applied Linguistics

Doctoral Thesis

Author: Jiří Semecký

Supervisor Prof. RNDr. Jan Hajič, Dr.
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské náměstí 25, 118 00 Prague 1

Department: Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské náměstí 25, 118 00 Praha 1

Opponents: RNDr. Pavel Krbec Ph.D.
NetCentrum, s. r. o.
Drtinova 10, Praha 5, 150 00

RNDr. Markéta Lopatková, Ph.D.
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské náměstí 25, 118 00 Praha 1

Abstract

Semantic analysis has become a bottleneck of many natural language applications. Machine translation, automatic question answering, dialog management, and others rely on high quality semantic analysis.

Verbs are central elements of clauses with strong influence on the realization of whole sentences. Therefore the semantic analysis of verbs plays a key role in the analysis of natural language. We believe that solid disambiguation of verb senses can boost the performance of many real-life applications.

In this thesis, we investigate the potential of statistical disambiguation of verb senses. Each verb occurrence can be described by diverse types of information. We investigate which information is worth considering when determining the sense of verbs. Different types of classification methods are tested with regard to the topic. In particular, we compared the Naïve Bayes classifier, decision trees, rule-based method, maximum entropy, and support vector machines. The proposed methods are thoroughly evaluated on two different Czech corpora, VALEVAL and the Prague Dependency Treebank. Significant improvement over the baseline is observed.

Declaration

I hereby declare that this thesis is my own work and where it draws on the work of others it is properly cited in the text.

Acknowledgments

First of all I would like to thank my parents for their continuous support during the years of my studies. I would also like to thank my wife for making my life so pleasant.

My supervisor Jan Hajič guided me in my research. I am also grateful to him for allowing me to study for one year at the Saarland University, Saarbrücken, Germany where I first got into lexical semantics.

I would like to express my special gratitude to the headquarters of the Institute of Formal and Applied Linguistics for providing an extraordinarily pleasant conditions for many young researchers.

This research would not be possible without financial support from the Faculty of Mathematics and Physics, the Institute of Formal and Applied Linguistics where I was employed and the Grant Agency of Charles University.

Finally, I would like to thank my colleagues Petr Podveský, Ondra Bojar, Petr Pajas, Zdeněk Žabokrtský, Markéta Lopatková, Nino Peterek, Jura Havelka, Honza Štepánek, Vašek Novák and Petr Němec for numerous discussions and for creating a nice working atmosphere at the department.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural language processing (NLP) research has already grown from the early phases of its life. Many tasks concerning the early stages of the linguistic analysis of written text, including lemmatization, morphological tagging and surface parsing, might today be considered sufficiently resolved for the mainstream NLP languages. Even if their development will probably further continue to improve, their current results are near to approaching the upper limits and they are already good enough for many practical applications.

The complex linguistic applications, including machine translation, question answering, dialog systems, information retrieval, and others however need deeper semantic analysis of text which is becoming the center of interest for current NLP research. This analysis tries to understand and describe not only the structure of text but also its meaning. But not all parts of speech are equally important for deep analysis.

Verbs have special roles in the analysis of text. From a syntactical point of view they are the central elements of clauses with direct influence on the presence and realization of other constituents. From a semantical point of view they are the bearer of events and their proper analysis is fundamental for the correct analysis of the rest of the sentence.

Moreover, verbs are also interesting from a linguistic perspective because they have the richest syntactical structure and also the highest level of ambiguity compared to other parts of speech.

Let us take a highly ambiguous Czech verb *dát* as an example. If we want to translate the verb into English, the most obvious translation will be *to give* as in the sentence:

Petr ***dal*** *Janě knihu. = Peter* ***gave*** *Jane a book.*

If we use the verb in combination with a reflexive particle *si* it changes the meaning of the sentence, and the verb needs to be translated as *put*:

Petr ***si dal*** *klíče do kapsy. = Peter* ***put*** *his keys in his pocket.*

Even with the same syntactical structure, we can get a completely different meaning which, again, translates differently:

Petr si dal Guinness do půllitru. = Peter ***ordered*** *a pint of Guinness.*

Needless to say, that when used in an idiomatic expression, the verb has a completely different translation:

Petr ***si*** *na tom* ***dal*** *záležet. = Peter* ***made*** *a point of it.*

Petr ***dal*** *na jeho slova. = Peter* ***took*** *what he said* ***into*** *account.*

Petr ***se dal*** *konečně dohromady. = Peter finally* ***got*** *better.*

As has been shown, the same Czech verb can be translated into different English verbs, depending on the sense in which it is used. Therefore, the correct assignation of the sense seems to be essential for the translation of the sentence. For other applications dealing with the semantic content of the text, it is naturally important too.

This work is dedicated to the process and methods of automatically choosing the proper sense of verbs in their given context, i.e. verb disambiguation[1] according to a certain definition of verb senses – lexicon.

Czech is one of the languages which are the center of study of the worldwide computational linguistic community. A significant reason for this is the fact that there is a large amount of high-quality linguistically annotated data. As there are only ten million Czech native speakers, other languages, mainly English, Chinese, French, Spanish, and Arabic definitely receive more attention because of the far larger number of target users. However, the Czech language surely has the highest ratio of linguistically annotated tokens per native speaker[2].

In our experiments we use two Czech corpora:

[1] *to disambiguate = to remove uncertainty of meaning from* (Oxford Dictionary)

[2] We state here this claim without precise proof, and assuming the exclusion of dead (or nearly dead) languages where the ration is (or approaches) infinity, even with a very limited corpus.

First, **VALEVAL**, a small but reliable corpus, containing a few thousand running verbs in contexts annotated by three annotators parallelly. The corpus was put together as a lexical sampling experiment for an existing valency lexicon, and contains sentences randomly selected from the Czech National Corpus. Only the selected verbs are annotated in the corpus. The sentences are not selected in any larger continuous blocks except for a small context attached to each annotated unit. Only the golden part of the corpus was taken into account in our experiments. This assured highly reliable labeling which had, however, low coverage and loose verb distribution.

Second, the tectogrammatical part of the **Prague Dependency Treebank 2.0**, a large corpus, containing almost 70,000 running verbs[3]. The tectogrammatical annotation layer describes many linguistic characteristics, including valency which was used as an approximation of verb senses as is explained later. Each sentence of the relevant portion of the Prague Dependency Treebank was annotated on the tectogrammatical layer by one annotator only, i.e. no parallel annotations were performed. Therefore, the quality of the valency annotation is not guaranteed to be as high as for the first corpus. On the other hand, the quantity highly exceeds VALEVAL and the distribution of verbs reflects the real distribution in Czech (newspaper) text.

Our disambiguation process can be simply described by a sequence of the following steps. First, we automatically linguistically analyzed the sentences containing the annotated verbs. Second, we created a vector of features for each annotated verb in the dataset, describing its context. We experimented with a large number of different features, a lot of attention was paid to the comparison of individual feature types. Third, the generated features were used in machine learning algorithms. Again, we experimented with several machine learning methods, including the Naïve Bayes classifier, decision trees, rule-based learning, support vector machines, and maximal entropy model. Finally, we evaluated the obtained results. In the evaluation section, we stated the results obtained by using all types of features separately, as well as using their different combinations. Also the difference in performance of individual classification methods are evaluated, as well as several other aspects.

[3] The number refers only to the portion annotated on the tectogrammatical layer.

1.1 Structure of the Thesis

Chapter 2 is a twenty-page introduction to machine learning methods which are later used in the work. You can skip it without missing out on anything unless you are going to implement the methods or are interested in the computational details.

Chapter 3 is short and presents an important shift in the thesis goal. You should read it even if you do not want to read much, while at the same time not losing the plot.

Chapter 4 is a short chapter describing different data resources used in the experiments. It is a good introduction to the data used, giving information which is referred to later in the work. If you are familiar with the data, reading this chapter will not provide you with much new information.

Chapter 5 presents the approaches of other authors solving similar problems or presenting a work which is relevant to the work presented in this thesis. Omitting this one will not affect your understanding of the thesis.

Chapter 6 describes the set of features which we used in our experiments. It is a relatively long chapter containing the main idea put forward in the thesis. It is where one should look if he or she is going to inquire into the implementational details.

Chapter 7 is the longest chapter giving all the evaluation results. It contains a lot of charts and tables with the outcomes seen from different perspectives. This chapter is the most interesting to read, especially if you are interested mainly in the quantitative results.

Chapters 8 summarizes and concludes the work. It does not provide any new information except for the future outlook.

Appendices are intended for those who are more concerned with the details of the work.

I wish you pleasant reading.

Chapter 2

Machine Learning Methods

2.1 Introduction

An incredibly large number of different machine learning methods are used in the computational linguistics for solving diverse types of (classification or regression) problems. Different methods are appropriate for different types of problems and the choice of the right one is crucial for solving the problem well. However, a complete comparison of machine learning methods is beyond the scope of this work.

We only introduce the theory behind classification machine learning methods which we later use in the work. Namely, we describe the Naïve Bayes classifier, different decision tree algorithms, support vector machines, and maximum entropy model. We also do not aim to provide in-depth analysis of the methods.

2.1.1 Classification Task

The term **classification task** is used for the task of labeling given objects with a predefined set of possible values. The classification task might be solved using machine learning methods, i.e. solved by algorithms automatically gained by computers. The task can also be solved differently, for instance by algorithms defined by human experts, or by the experts themselves. However, we are not going to discuss this topic here any further and we will focus on the machine learning methods.

Machine learning classification methods generally derive the knowledge of how to classify (i.e. the classification algorithm) from the **training data** given them in advance. There are two basic types – supervised methods and unsupervised methods.

The **supervised methods** derive the knowledge from labeled (classified) data. They are trained on a portion of data for which the desired result is known in advance. The labeling of training data is believed to be correct – they are usually gained by measuring some quantity of real objects or by another reliable source (e.g. a human expert). The latter case is common in computational linguistics.

The **unsupervised methods** derive the knowledge from unlabeled data, i.e. data for which the correct classification is not known in advance. Of course, learning from unlabeled data is much harder and the methods are often more complicated. On the other hand, the unlabeled data are easier to obtain and they might often be available in a significantly higher quantity. Supervised and unsupervised methods are also often combined, where usually the supervised method trained on a relatively small amount of labeled data provides the basic structure of the knowledge and the unsupervised method is subsequently used to tune the weights by robust statistics.

In this work, we will be using only supervised methods, and unless stated differently, the term *machine learning methods* will refer to supervised machine learning methods.

2.1.2 Methodology

The machine learning experiment usually follows this methodology:

- The labeled data divided into two parts – training and testing data set.
- The training data set is used for training the classification method, as mentioned above.
- The testing data are left aside during the training phase. Later, they are classified by the induced algorithm in the same manner as the unlabeled data (the labels are ignored), and the result classification is compared to the original, assumably correct, labels to measure the reliability of the method.
- The reliability is expressed in terms of accuracy, precision, recall, f-measure or other metrics.

The portion between the training and testing data is not always the same and it depends on the desired quality of the results. The more data are used for the training phase, the higher expected quality of the trained algorithm. The more data are spared for the testing phase, the more reliable will the evaluation be. Usually, the testing data have between 10% to 30% of the whole portion of the data.

In some cases, the training data are further divided into more sub-parts, one part is used for the training of the algorithm (training data), and the others are used for subsequent tuning of different parameters (held-out data).

In reality, the amount of training data is often considerably low because of the nature of the data or just because they are too difficult to obtain. We can not afford to lose much data from the training data set, but the low amount of data in the testing data set would hurt the evaluation. In such cases, the cross-validation might come into play.

In **cross-validation** we split the data into n parts. We run the training phase n-times, for each n-th, test the method on the selected n-th, and we train it on the remaining data. Finally, we make the overall evaluation as a combination of the obtained results. Using this trick, we always train using a high portion of the data (90% for ten folds), but the evaluation is computed over all the annotated data. The downside of cross-validation is the n-times more time required.

2.1.3 Representation of Data

The classified objects in the field of computational linguistics are usually linguistic objects, such as words, sentences, phonemes, etc. On behalf of computer processing, the objects are described in a formal, computer understandable, way. Without the loss of generality, we will choose **vectors** of atomic values as this formal representation. These values will be referred to as **features**.

Features can bear numerical (ordinal or floating-point) or categorial values. The main difference is that for the numerical features there is an ordering of their possible values, for the categorial features there is not.

An example of a numerical value from natural language processing field is “the number of words in sentence”, it can be ordered naturally. An example of a categorial value is case (for Czech, there are seven cases: $case \in \{1, 2, 3, 4, 5, 6, 7\}$). Even if the value can be expressed as a number, there is no natural ordering among the possible values. A special case of a categorical feature is a boolean feature (binary feature) which can bear only two values (*true* and *false* or 1 and 0).

The description of the linguistic objects as vectors of values as used in our experiment is thoroughly discussed in Chapter 6.

Each object from the labeled data is accompanied by a label (classification) which is a categorial feature, representing the class (category) to which the object belongs.

2.2 Naïve Bayes Classifier

Naïve Bayes Classifier [Langley et al., 1992] is a simple probabilistic classifier based on probability models and on an assumption that features are independent of each other. This assumption does not often hold in reality (hence *naïve*). The probability model is derived using Bayes' theorem (hence *Bayes*).

Naïve Bayes classifier computes the probability that an object represented by a vector of features belongs to a given class separately for each feature in the vector and computes the overall probability as if the features were mutually independent. In practical applications, parameters for the Naïve Bayes classifier are often estimated using the maximum likelihood estimation [Aldrich, 1997].

2.2.1 The Probabilistic Model

The probabilistic model for the classifier is a conditional model

$$p(c \mid f_1, f_2, \dots f_n)$$

with a dependent class variable c and independent feature variables
– f_1 through f_n.

Using Bayes' theorem

$$p(c \mid f_1, f_2, \dots f_n) = \frac{p(c) \cdot p(f_1, f_2, \dots f_n \mid c)}{p(f_1, f_2, \dots f_n)}.$$

When classifying an unlabeled object, we select the class for which the probability is maximal given the features. Since the features are fixed for the given sample, the denominator is constant and therefore:

$$\begin{aligned} \arg\max_{c \in C} p(c \mid f_1, f_2, \dots f_n) &= \arg\max_{c \in C} \left(\frac{p(c) \cdot p(f_1, f_2, \dots f_n \mid c)}{p(f_1, f_2, \dots f_n)} \right) \\ &= \arg\max_{c \in C} \Big(p(c, f_1, f_2, \dots f_n) \Big). \end{aligned}$$

Using the independence assumption

$$p(f_i \mid f_j, c) = p(f_i \mid c) \quad \text{, for } i \neq j$$

we can rewrite the inside of the *argmax* function as follows

$$\begin{aligned}
p(c, f_1, f_2, \dots f_n) &= p(c) \cdot p(f_1, \dots f_n \mid c) \\
&= p(c) \cdot p(f_1 \mid c) \cdot p(f_2, \dots f_n \mid c, f_1) \\
&= p(c) \cdot p(f_1 \mid c) \cdot p(f_2, \dots f_n \mid c) \\
&= \dots \\
&= p(c) \cdot p(f_1 \mid c) \cdot p(f_2 \mid c) \cdot p(f_3 \mid c) \cdot \dots \\
&= p(c) \cdot \prod_{i=1}^{n} p(f_i \mid c)
\end{aligned}$$

The conditional distribution over the class variable is:

$$p(c \mid f_1, f_2, \dots f_n) = \frac{1}{Z} \cdot p(c) \cdot \prod_{i=1}^{n} p(f_i \mid c)$$

where

$$Z = p(f_1, f_2, \dots, f_n)$$

is constant given the vector of features (description of the object).

2.2.2 Training

To train the classifier, we need to estimate the parameters of the probability models. Due to the independent features assumption, it is sufficient to estimate the parameters for each feature separately, for example using the maximum likelihood estimation:

$$p(f_i|c) = \frac{p(f_i, c)}{p(c)} := \frac{count(f_i, c)}{count(c)}$$

where the function count gives the number of samples in the training data with the corresponding values of the features and the class.

2.2.3 Classification

To classify an unknown case, we need a decision rule for selecting a result class. A common rule is the maximum a posteriori decision rule (picking the hypothesis that is most probable):

$$NBC(f_1, \dots, f_n) = \arg\max_{c} p(C = c) \prod_{i=1}^{n} p(F_i = f_i \mid C = c)$$

In spite of its very simple design and the fact that the independence assumption is often strongly violated, the Naïve Bayes classifier has several properties that make it surprisingly useful in practice and it often works much better in many complex real-world situations than it might be expected. However, it is not one of those classifications whose performance is expected to be the cutting edge.

2.3 Decision Tree

In general, decision tree algorithms [Buntine, 1993] are classification models based on a set of decisions ordered in a tree-like structure. The classifier is described by a directed acyclic graph in form of a tree (in the sense of graph theory). Each inner node of the graph corresponds to a feature, edges represent possible values of the feature in their parent node. Leaves represent the predicted classes given the values of the features represented by the path from the root.

There are two basic types of decision trees:

- **Classification tree**: classification method whose predicted class is a categorical variable, i.e. the outcome is chosen from a finite number of possible values (e.g. *true/false*, verb sense, ...)
- **Regression tree**: classification method whose predicted value is a real number value (e.g. predicted price of a stock, temperature prediction, ...)

In this work we are using merely classification trees.

The classification decision tree predicts class of an object from a set of possible outcomes using vector of features describing the object.

Decision trees have several advantages which makes them popular for data classification. First, they do not need any data normalization or creation of dummy variables. Second, decision trees are white-box model, i.e. behavior of the model could be easily explained by the boolean logic directly resulting from the tree. On the contrary, methods like maximum entropy model, neural networks or support vector machines are black-box models. Third, it is possible to validate a model using statistical tests. And finally, the model is robust and relatively fast even on large training data.

Figure 2.1 shows an example on an decision tree, which corresponds to algorithm given in Algorithm 1.

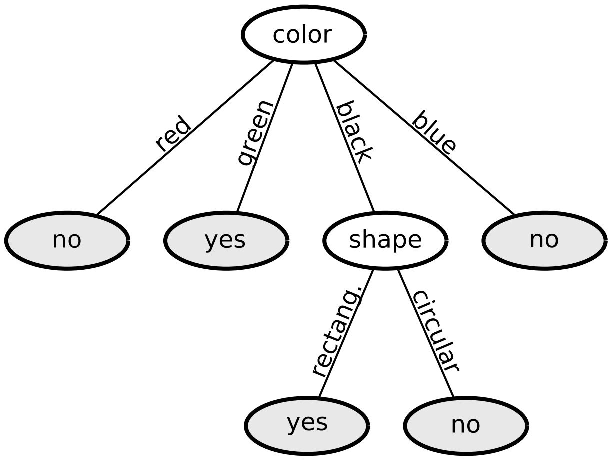


Figure 2.1: Example of a decision tree

2.3.1 Training

In the training phase, the decision tree algorithm is automatically induced from labeled data. Decision tree is a generic algorithm, concrete instances of the algorithm differ in the way how they induce the tree.

We will assume that the data (object descriptions) come in the form:

$$(c, \mathbf{F}) = (c, f_1, f_2, \ldots, f_n)$$

where $c \in C$ is the class variable, and f_1 through f_n are the feature variables.

The induction of the tree combines greedy selection of features with the *divide and conquer* principle, so the algorithm consists of two phases:

- In the first (greedy) phase, the algorithm computes conditional frequency distributions of the classes given the provided features. The distributions are evaluated using some measure (depending on the concrete algorithm) and the feature with best performance (maximizing or minimizing the measure) is selected as the next test feature and assigned to the created node.

- In the second phase, the *divide and conquer* principle comes in play. The data are divided into groups according to the possible values of the chosen feature, and the procedure applies recursively for each group separately. Each branch of the tree has less training data than the previous one.

Algorithm 1 Algorithm represented by decision tree in Figure 2.1

```
if color == red then
  return no
else if color == green then
  return yes
else if color == black then
  if shape == rectangular then
    return yes
  else if shape == circular then
    return no
  end if
else if color == blue then
  return no
end if
```

The recursion stops if all data instances of the group belong to a single class or if there is no more discriminative feature to divide the data or if some other stopping criteria holds.

Algorithm 2 shows the general decision three induction algorithm.

2.3.2 Classifying

In the later phase, the tree obtained from the learning phase is used to predict classes of unlabeled objects. The tree is browsed from the root downwards following the path corresponding to the feature values of the object. The leaf which is reached assigns the class to the object.

2.3.3 Measures

In this section, we introduce two measures which are commonly used by decision tree algorithms for different purposes, e.g. to choose the best feature to split the data, to decide whether stop or continue with the induction, etc.

We use $c \in C$ for the class variable. Further, let $f_N(c)$ be the relative frequency of objects belonging to a class c in a tree node N (remember that each node operats with a different subset of data).

Gini impurity is a measure used by an algorithm called **CART** (Classification and Regression Trees) [Breiman et al., 1993]. The measure shows how much the data differs within one class. It is equal to zero if all data belong to the same class. The value of the measure is calculated as squared

Algorithm 2 InduceDecisionTree ($Samples$: array of feature vectors with assigned classes, $Features$: feature list)

```
Create a new node $new\_node$
if (all samples belong to one class) then
  Assign this class to the $new\_node$ and return it as a leaf
else if ($Features$ is empty) then
  Assign some label (e.g. the label of the most common class value) to the
  $new\_node$ and return it as a leaf
else if ($Samples$ is empty) then
  Assign some label (e.g. the label of the most common class value) to the
  $new\_node$ and return it as a leaf
else
  $F$ := select feature from $Features$ maximizing the measure
  (depending on concrete algorithm)
  Set the decision attribute for $new\_node$ to $F$
  for $f \in valuesOf(F)$ do
    $S := \{s \in Samples|$ the value of the feature $F$ of the sample $s$ is $f\}$
    $T$ := IndurceDecisionTree($S$, $Features \setminus F$)
    Add new tree edge bellow $new\_node$ corresponding to test $F == f$
    Connect this edge to $T$
  end for
end if
Return $new\_node$
```

probabilities of the relative frequencies of all the classes subtracted from one:

$$I_G = 1 - \sum_{c \in C} f_N(c)^2$$

Entropy. Many decision tree algorithms, including ID3, C4.5, and C5.0, are based on the notion of entropy used in information theory.

$$I_E = -\sum_{c \in C} f_N(c) \cdot log_2 f_N(c)$$

In the following paragraphs, we introduce a few particular decision tree algorithms.

2.3.4 ID3 Algorithm

Iterative Dichotomiser 3 (ID3) [Quinlan, 1986], [Quinlan, 1996] algorithm is a decision tree algorithm for boolean classification. It is based on the general algorithm stated in section 2.3.1 and for building the tree, the algorithm uses the entropy measure.

Let $p_{samples}(c_i)$ be the probability of class c_i in a set of labeled data – *samples*. Then the **entropy** (measure of the amount of uncertainty) of the *samples* is

$$I_E(samples) = -\sum_{c \in C} p_{samples}(c_i) \cdot log_2\big(p_{samples}(c_i)\big)$$

Let us assume that $values(f)$ is defined as a set of all possible value of the feature f, and $samples_{[f=v]}$ is the subset of *samples* for which the feature f has value v. Than we define the **information gain** as:

$$I_G(samples, f) = I_E(samples) - \sum_{v \in values(f)} \frac{|samples_{[f=v]}|}{|samples|} \cdot I_E(samples_{[f=v]})$$

For each node, the ID3 algorithm chooses the feature with the highest information gain.

The Algorithm 3 shows the pseudocode of the ID3 algorithm.

Algorithm 3 InduceDecisionTree ($Samples$: array of feature vectors with assigned classes, $Features$: feature list)

```
Create a new node new_node
if (all samples are positive) then
  Assign the positive value to the new_node and return it
else if (all samples are negative) then
  Assign the negative value to the new_node and return it
else if ($Features$ is empty) then
  Assign the label of the most common class value to the new_node and
  return it as a leaf
else if ($Samples$ is empty) then
  Assign the label of the most common class value to the new_node and
  return it as a leaf
else
  $F$ := feature with the biggest information gain
  Set the decision attribute for new_node to $F$
  for $f \in valuesOf(F)$ do
    $S := \{s \in Samples|$ the value of the feature $F$ of the sample $s$ is $f\}$
    $T$ := IndurceDecisionTree($S$, $Features \setminus F$)
    Add new tree edge bellow new_node corresponding to test $F == f$
    Connect this edge to $T$
  end for
end if
Return new_node
```

2.3.5 C4.5 Algorithm

C4.5 [Quinlan, 1993] is a decision tree algorithm based on the ID3 algorithm, originally designed by Ross Quinlan. It contains several improvements over the ID3 algorithm stated in the following list:

- **Missing values** of features are allowed. They are ignored in the induction phase. In the classification phase the missing values are interpolated from the labeled data.

- C4.5 makes it possible to use **continuous feature values**, whereas ID3 works with categorical values only.

- C4.5 supports **tree pruning** by dividing the learning data into training and validation set. Each newly induced branching is tested on the validation set and if the result is worse than the original result, the branching is ignored and a single node corresponding to the most frequent value is used instead.

- **Reduced error pruning** tries to replace a subtree with its most common class value. The reduced error pruning is reflected in the result tree only if it brings classification improvement on the validation data.
- C4.5 supports **subtree rising** – the algorithm tries to replace a part of tree with its most common subtree, so a common subtree is raised several levels up. The subtree rising takes place only if it brings improvement on the validation data.

The division criteria of the C4.5 also differs from the ID3. Instead of using information gain, it uses **information gain ratio** which is defined as follows:

$$I_{GR}(samples, f) = \frac{I_G(samples, f)}{\sum_{v \in values(f)} (\frac{|samples_{f=v}|}{|samples|} \cdot log_2 \frac{|samples_{f=v}|}{|samples|})}$$

The C4.5 algorithm stops branching if all samples of the data corresponding to node belong to the same class or if the subsequent branching does not bring any further improvement on the validation data set.

2.3.6 C5.0 Algorithm

The C5.0 [Quinlan, 2002] algorithm is a new version of the C4.5 algorithm, which is, however, not published but distributed as a commercial tool. It uses similar induction of decision tree as the C4.5 algorithm, but the generation is more effective.

The C5.0 could generate more reliable classifiers thanks to **boosting** which combines different classifiers (dividing the labeled data into more groups). In the classification phase, the resulting class is achieved by voting.

Over the C4.5, the C5.0 contains other improvements including different misclassification costs for different classes (and in the recently (2007) published version[1] even different data instances) and automatic winnowing of unused attributes.

The C5.0 algorithm is implemented in the See5/C5.0 toolkit[2] [Quinlan, 2002] developed by Rulequest research in Autralia.

[1] http://www.rulequest.com/r204.html

[2] http://www.rulequest.com/see5-info.html

There is not much to say about the concrete implementation, because the algorithms, as of a commercial product, are not publicly known by the time of writing this thesis.

2.4 Rule-sets

The C5.0 toolkit [Quinlan, 2002], [Buntine, 1990] allows for inferring sets of decision rules from the decision trees. It constructs a single rule for each leaf in the tree.

Subsequently, the algorithm applies **rule post-pruning** which generalizes the rules by removing different rule conditions when such simplification does not hurt the classification.

The rules are independent of each other and therefore their conditions can overlap in which case the classification is achieved by using the rule with the highest predicted preciseness.

Because of the way in which the rules are constructed, the rules tend to be highly correlated with the C5.0 decision trees, however the classifier might differ in certain cases, and the authors claim that the rule-sets usually perform better than the decision trees.

2.5 Support Vector Machine

Support vector machine (SVM) is a supervised kernel-based method for vector classification. It applies a linear separator on a kernel-modified feature space. The key trick is the kernel-transformation of a non-linear problem into a linear one – the original feature space is modified by a non-linear kernel function and then **linearly** divided into two subspaces by a hyperplane.

The method for linear space division was introduces in the 1960's by Vladimir Vapnik, it was however not until 1992 when [Boser et al., 1992] proposed the kernel modification allowing usage of the methods for non-linear classification. Recently, in 1995, [Cortes and Vapnik, 1995] introduces soft-margin modification, which lets SVM to be used for non-separable problems. Where the feature space can not be separated into two classes, the algorithm chooses a separator that splits the examples as cleanly as possible, while still leaving mislabeled examples. This modification made the SVM a popular and widely used method and the term “Support Vector Machines” became a notion in computer science.

2.5.1 Linear Classification

Suppose we have instances of training data described by the feature vectors

$$(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}) \in \chi$$

classified into two classes. Function c defines the classification:

$$c(i) \in \{+1, -1\} \text{ , for } i \in 1, \ldots, n.$$

The feature space χ must be a space with inner product, a common example of such space is the $\mathbf{R^n}$ space.

Our aim is to linearly separate the space by a hyperplane (a space of a dimension smaller by one than the dimension of the original space). We will use the term **margin** for the distance of such hyperplane flom the closest data-point (of either class). From all separating hyperplanes we choose the one with the maximal margin. This hyperplane is referred to as **maximum-margin hyperplane** or **optimal hyperplane**. There are several reasons for choosing the hyperplane this way. It feels intuitively safest, because if we have made an error in locating the boundary, it gives us the biggest possible fall-back. Also, if the boundary is correct, we allow the biggest possible error in input values. The Vapnik-Chervonenski's theory suggests that this is a good choice. And finally, this works well empirically.

Let us consider an $\mathbf{R^2}$ example. Than the the hyperplane in Figure 2.2 is an optimal hyperplane, while the hyperplane in Figure 2.3 is a hyperplane still separating the vectors, yet not optimally.

Formally, each hyperplane can be described[3] by a vector $\mathbf{w}$ and a number b:

$$\mathbf{w} \cdot \mathbf{x} - b = 0.$$

The margin of such hyperplane is defined as a set of points $\mathbf{x}$ for which the following equation holds:

$$\mathbf{w} \cdot \mathbf{x} - b \in (-1, 1)$$

Note that we can adjust the size of the margin by multiplying the vector $\mathbf{w}$ by a scalar value, but the b-value also has to be changed accordingly.

We search for such hyperplane that correctly classify all training data, which is:

$$\mathbf{w} \cdot \mathbf{x_i} - b \geq 1 \text{ for all } c(i) = +1 \text{ (positive samples)}$$

[3] The symbol "$\cdot$" (in $\mathbf{w} \cdot \mathbf{x}$) is used for inner product

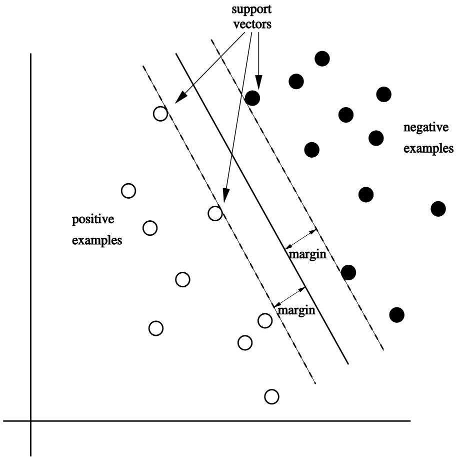


Figure 2.2: SVM: optimal hyperplane separating sample data

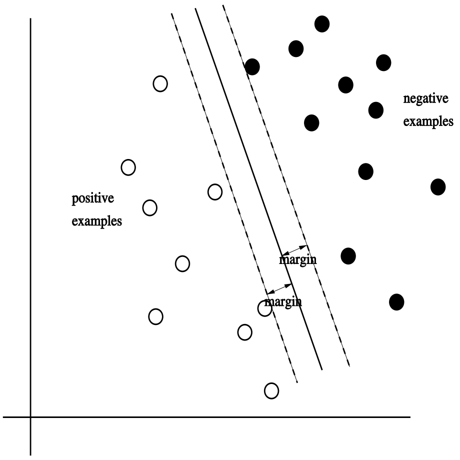


Figure 2.3: SVM: non-optimal hyperplane separating sample data

$$\mathbf{w} \cdot \mathbf{x_i} - b \leq 1 \text{ for all } c(i) = -1 \text{ (negative samples) .}$$

The equation can be merged as:

$$c(i) \cdot (\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1 \text{ for } i \in < 1, \ldots, n > .$$

While holding this condition, we want to maximize the margin.

It can be seen from Figure 2.2 that the values of **w** and b do not depend on the “inner” data-points, but only on those which are situated on the margin. Those points (vectors) are called **support vectors**. For all support vectors, the following equality holds:

$$abs(\mathbf{w} \cdot \mathbf{x_i} - b) = 1.$$

Let us consider two support vectors – x_1 from the positive set, and x_2 from the negative set:

$$\mathbf{w} \cdot \mathbf{x_1} - b = 1,$$

$$\mathbf{w} \cdot \mathbf{x_2} - b = -1.$$

Combining these equations we get:

$$\mathbf{w} \cdot (\mathbf{x_1} - \mathbf{x_2}) = 2$$

$$\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot (\mathbf{x_1} - \mathbf{x_2}) = \frac{2}{\|\mathbf{w}\|}$$

When x_1 and x_2 are support vectors from positive and negative dataset, respectively, the value $\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot (\mathbf{x_1} - \mathbf{x_2})$ is the margin, measured perpendicularly to the hyperplane. As we want to maximize the margin, we will minimize the $\|\mathbf{w}\|$ subject to the constraints:

$$c(i) \cdot (\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1 \text{ for } i \in < 1, \ldots n >$$

This is an optimization task which can be solved using Lagrange multipliers and leads to a quadratic programming problem which can be solved.

2.5.2 Non-linear Classification

Up to now, we have solved only linear classification task. However, sometimes feature space can not be satisfactorily separated by a hyperplane because of the nature of the data. In such cases, non-linear separation would be much more accurate. In this section we give a brief introduction into non-linear separation with SVM.

A straight-forward solution of non-linear separation is a direct description of a more complex separator in the feature space of the data. However, the kernel-trick used in SVM handles this problem differently – by a non-linear transformation (τ) of the data into a different feature space where the hyperplane can already be used for separating the data appropriately.

The algorithms for linear classification described in the previous section uses only one operation on the feature space – the inner products. No other assumption except for the existence of the inner product operation was posed on the feature space. Therefore, when using the kernel transformation, the inner-product operation must be defined on the space of the tranformation results.

As the vectors in the transformed space occur the equations only as arguments of inner products, we can join the transformation τ and the inner product operation into a single function – the **kernel function**:

$$k(\mathbf{x1}, \mathbf{x2}) = \tau(\mathbf{x1}) \cdot \tau(\mathbf{x2}).$$

Technically, we only substitute the inner products by a kernel function in the equations designed for the linear model, and the kernel function adds the non-linearity into the model.

The kernel function may also transform data into a higher dimension space where the linear separation might be feasible. An example of such transformation is function:

$$\tau(x_1, x_2) = \big(x_1, x_2, \sqrt{x_1^2, x_2^2}\big)$$

from R^2 into R^3.

The following list gives examples of commonly used kernel functions:

- Polynomial functions: $k(x_1, x_2) = ((x_1^\top \cdot x_2) + \theta)$
- Sigmoid: $k(x_1, x_2) = tanh(\kappa(x_1^\top \cdot x_2) + \theta)$
- Gaussian radial basis function: $k(x_1, x_2) = exp(-\frac{\|x_1 - x_2\|}{2\sigma^2})$

The computationally difficult problem in SVM is finding the appropriate parameters for the kernel functions.

2.6 Maximum Entropy

The principle of maximum entropy [Berger et al., 1996] is a method for determining a unique epistemic probability distribution making use of prior information. It is based on entropy. The notion of entropy originally comes from physics where it was defined as a state function of a thermodynamic system denoting measure of disorder of the system. In statistics, the notion of entropy was introduced by Claude E. Shannon as a measure of uncertainty of a probability distribution:

$$H(p) = -\sum_i p_i \cdot log(p_i)$$

The principle of maximum entropy is based on the principle of Occam's Razor – it uses as much information as available for determining constraints of an unknown probability distribution, but it does not make up any further information. It rather prefers the most uniform model. This is the model with minimum additional information (i.e. with maximum entropy) while being consistent with the known constraints.

Here comes an example of the model:
Let us consider an experiment with 10 possible outcomes $(A \dots J)$. If we have no knowledge about what the outcome can be, according to the principle of maximum entropy, we should use the probability distribution giving the same chance $(p = 0.1)$ to all possible outcomes. If we add a knowledge claiming that the outcome A has higher probability, say $p(A) = 0.4$, the probability distribution should reflect this knowledge, giving the outcome A the corresponding outcome, while keeping the rest of the outcome uniform probability $(P(notA) = 0.067)$.

In general, the principle of maximum entropy can be used for estimating any probability distribution. In this work, we will use it for classification, and we concentrate on the conditional distribution from learning data.

2.6.1 Constraints

In the maximum entropy model, we induce constraints on the conditional distribution. Each constraint expresses a characteristic (feature) of the training data that should also hold for the testing data. We describe the features as real-value functions, $f_i(d, c)$ will stand for the value of i-th feature for the given data sample d and class c. We can than restrict the model distribution so that the expected value of this feature corresponds to the empiricaly measured value in the training data:

$$\frac{1}{\mid D \mid} \sum_{d \in D} f_i(d, c(d)) = \sum_{c,d} P(d) \sum_{c} P(c|d)$$

where D is the data set, and the function $c(d)$ assigns the classification to a data instance according to the training data.

Further, we are not interested in the distribution of the data instances, so we use our training data to approximate it. Thus we can rewrite the equation as:

$$\frac{1}{\mid D \mid} \sum_{d \in D} f_i(d, c(d)) = \sum_{d \in D} \sum_{c} P(c, d)$$

When using maximum entropy, the first step is to identify a set of features which will be useful for classification. Then, for each feature, we measure its expected value over the training data and take this to be a constraint for the model distribution. We have done those two steps so far.

2.6.2 From Constraints to Model

Given the constraints described in the previous paragraphs, there is a distribution that has the maximum entropy [Della Pietra et al., 1997]. Up to now, however, we do not know how the distribution looks like. Maximizing the entropy function $H(p)$ over some held-out data in the presence of constraints $constr_i = v_i$ can be reformulated as maximizing functions

$$H(p) - \sum_h \lambda_i \cdot (constr_i - v_i).$$

If Exp_{train} denotes the expectation over the training data, and Exp_p the expectation over the held-out data, this can be reformulated as maximizing

$$H(p) - \sum_h \lambda_i \cdot (Exp_p(f_i(c,d)) - Exp_{train}(f_i(c,d)))$$

$$H(p) - \sum_h \lambda_i \cdot (\frac{1}{\mid D \mid} \sum_{d \in D} f_i(d, c(d)) - \sum_{d \in D} \sum_c P(c,d)$$

over the parameters λ_i and the distribution p.

Further, we add a constraint to ensure that probabilities add to one.

$$\sum_{c \in C} P(c|d) = 1$$

We end up with a system of constraining equations, using one variable for each feature plus one from definition of the probability. For solving this system we can use Lagrangian Multipliers, a method for finding local maxima on a restricted subspace. The idea of Lagrangian Multipliers is to find a points in the subspace where all the partial derivation equals to zero and pick up the one where function has a global maximum. We state without further evidence that after solving the Lagrangian Multipliers, we get to a solution describing the probability distribution solely in terms of λs:

$$p(c|d) = \frac{1}{Z(d)} \cdot exp(\sum_i \lambda_i f_i(d,c))$$

where $f_i(d,c)$ is a i-th feature, λ_i is i-th parameter to be estimated and $Z(d)$ is a normalization factors to make the probability sum to one.

$$Z(d) = \sum_c exp(\sum_i \lambda_i f_i(d,c))$$

2.6.3 Acquiring the Parameters

In many classification tasks, the parameters are derived from the labeled training data. It is guaranteed that the likelihood surface is convex, having a single global maximum and no local maxima. This suggests a possible approach for finding the local maximum – we can guess the initial values of the parameters, and iteratively improve the function by climbing on the convex likelihood space. Since there is no local maxima, this will converge to the maximum likelihood solution for exponential models, which will also be the global solution of the maximum entropy model.

Different algorithms can be used. For instance, **Generalized Iterative Scaling** [Darroch and Ratcliff, 1972] (GIS) and the **Improved Iterative Scaling** [Pietra et al., 1997] (IIS) are two high-climbing algorithms commonly used for calculating the parameters.

The ISS performs the high-climbing in the logarithmic likelihood space. Given the training data D, the likelihood of the logarithmic model can be formulated as:

$$\begin{aligned}
l(\Lambda|D) &= log\big(\sum_{d\in D} P_\Lambda(c(d) \mid d)\big) \\
&= \sum_{d\in D} log\big(P_\Lambda(c(d) \mid d))\big) \\
&= \sum_{d\in D} log\big(\frac{1}{Z(d)^\Lambda} exp(\sum_i \lambda_i^\Lambda f_i(d,c))\big) \\
&= \sum_{d\in D}\sum_i \lambda_i^\Lambda f_i(d,c) - \sum_{d\in D} log \sum_c exp\big(\sum_i \lambda_i^\Lambda f_i(d,c)\big)
\end{aligned}$$

If the IIS finds a more likely set of parameters the likelihood of the original model increases as well because of convexity of the logarithmic function. This process iterates to the global maximum of both the logarithmic and the original maximum-entropy models. In each step the IIS algorithm computes the expected values of the current model distribution, and modifies the lambda parameters adequately.

Chapter 3

Word Senses

In this chapter, we show that what we are going to disambiguate in this work are actually not senses of verbs but their valency frames. We explain why we are using this approximation and show that under a specific assumption it does not really matter so much.

We have worked with two different lexicons, namely VALLEX, and PDT-VALLEX.

For building a statistical word sense disambiguation system, two types of data resources are needed – a lexicon defining word senses and a corpus annotated with the senses of this lexicon.

As far as we know, there is no Czech corpus of reasonable size annotated with senses of verbs according to a reliable lexicon. The WordNet is one attempt to define senses of words, however it is primarily oriented to nouns (though it contains other parts of speech too). Moreover, at the time this work was started, there was no Czech corpus of sufficient size annotated with the WordNet senses.

Because of this, we have decided to modify the task slightly by approximating verb senses with verb valency frames. Valency is a property of verbs which correlates with the senses to a certain extent, it is formally well defined and there are lexical resources of sufficient size available describing and using verb valency. In the following paragraphs, we point out that in our choice of valency frame lexicons, the correlation between frames and senses is relatively high.

3.1 Valency

Valency [Panevová, 1974], [Panevová, 1980], [Panevová, 1994] is the ability of a lexical item to combine with another lexical items in syntactical structures. The valency is defined for four different parts of speech — verbs, substantives, adjectives and adverbs. There is no doubt that the valency of verbs is the most differentiated and therefore the most interesting for studying. In this work we are only concerned with verb valency, leaving the valency of other parts of speech aside.

Here we mention a few alternative definitions of linguistic valency from different sources:

- **The American Heritage Dictionary of the English Language, Fourth Edition.** ([Kaethe, 2000]):
  Valency is the number of arguments that a lexical item, especially a verb, can combine with to make a syntactically well-formed sentence, often along with a description of the categories of those constituents. Intransitive verbs (appear, arrive) have a valence of one—the subject; some transitive verbs (paint, touch), two—the subject and direct object; other transitive verbs (ask, give), three—the subject, direct object, and indirect object.

- **Panevová** ([Petr Karlík, 2002]):
  Počet a povaha míst (argumentů), které na sebe sloveso (popř. jiný slovní druh) váže jako pozice obligatorní n. potenciální.

  The number and nature of the positions (arguments) which the verb (or a different part of speech) requires as obligatory or potential positions.

- **Žabokrtský** ([Žabokrstký, 2004]):
  Valency is a property of language units reflecting their combinatorial potential in language utterances.

- **Wikipedia.org**:
  In linguistics, **valency** or valence refers to the capacity of a verb to take a specific number and type of arguments.

Valency is described in terms of **valency frames** which defines the ability of the given lexical item to syntactically combine with other lexical item. If a verb can combine with lexical items in different manners, we say that the verb can occur in more different frames, or simply that the verb has a certain number of frames.

From a technial point of view a valency frame is usually described by a central lexical item (predicate, frame evoking element, ...) and a list of participants of the frame (arguments, frame elements, ...) corresponding to individual lexical items linked to the central element described by their linguistic (usually morphological and syntactical) characteristics and semantic labels. Different configurations of participants imply different valency frames. The participants are further categorized in different ways, depending on the concrete valency theory (e.g. usually distinguishing the level of obligatoriness).

3.2 Valency frames vs. senses

There is a general, many-to-many relation between the senses of a lexical item and its valency frames, i.e. a valency frame might correspond to different senses, in addition a particular sense can be realized by different valency frames.

We demonstrate this statement in the following Czech examples:

- A single valency frame can correspond to several senses:
  The verb *chovat* can bear a syntactical valency frame containing actor in nominative and patient in accousative (*ACT.1 PAT.4*). This frame corresponds to two different senses of the verb, namely - *cuddle* (*chovat dítě = cuddle a baby*) and *breed* (*chovat králíky = breed rabbits*).

- A single verb sense can be represented by different valency frames:
  The following sentences can be considered to have the same meaning, however the valency frames differs[1]:
  Naložit vůz senem.
  Load the cart with the hay.
  Naložit seno na vůz.
  Load the hay on the cart.

Despite this fact, there is no doubt that the valency is on a certain level of correlation with the meaning of the verb. We leave this claim without an exact proof and rely on the linguistic intuition of the reader to quantify the extent of the correlation.

More about verb Czech valency could be found in [Panevová, 1974] or [Žabokrstký, 2004].

3.3 Approximation of senses

We also showed that the valency frames, as defined in the previous chapter, do not correspond unambiguously to senses of verbs.

The valency lexicons built at the Institute of Formal and Applied Linguistics in Prague – VALLEX and PDT-VALLEX (introduced in Section 4.2) – are, however, different from the general definition in this point: the **clearly different senses of a verb with equal valency frames are distinguished in the lexicon**. The following examples demonstrate this statement:

[1] In some theories these participant configurations might be represented as alternations of the same valency frame.

VALLEX:

- **Frame** $\boxed{\textbf{1}}$: **ACT**$_1$ **PAT**$_4$
  absolvovat studium
  graduate from a place

- **Frame** $\boxed{\textbf{2}}$: **ACT**$_1$ **PAT**$_4$
  absolvovat operaci
  undergo an operation

PDT-VALLEX:

- **Frame** $\boxed{\textbf{v-w1184f1}}$: **ACT**$_1$ **PAT**$_4$
  chová prasata na farmě.LOC
  He breeds pigs on the farm.

$\vdots$

- **Frame** $\boxed{\textbf{v-w1184f4}}$: **ACT**$_1$ **PAT**$_4$
  chová dítě v náručí.LOC
  He cuddles the child in his arms.

When the difference in the meaning was not clear, frames did not have to be differentiated which corresponds to the uncertainty in the sense distinction.

From this perspective, **verb sense** (without any precise definition) **is a function of frames** (in VALLEX and PDT-VALLEX). The frame distinction in these lexicons is in fact driven by the combination of the valency and sense characteristics. Therefore these frames can be used as a suitable approximation of senses.

For the automatic assignment of word senses we need lexicon containing formal definitions of senses. As already suggested above, instead of using such lexicons we are using lexicons of valency frames which take senses distinction into account.

Chapter 4

Data resources

In this chapter, we describe the data which we used or referred to in the experiments discussed in the thesis. First, we introduce the Functional Generative Description, which is a theoretical base for most of the described data sources. Next, we present two valency lexicons together with two corresponding corpora. The lexicons define the senses of verbs and the corpora use those lexicons to annotate the verbs. We use the annotations as the training and the testing data in an experiment described later.

4.1 Functional Generative Description

Functional Generative Description [Sgall et al., 1986], is a linguistic framework developed by Prof. Sgall in the 1960's and motivated by the Prague Linguistic Circle, a linguistic working group which wan founded in 1926.

The Functional Generative Description (FGD) describes language on different layers where adjacent layers are related in the way that elements of the upper layer are functions of elements of the lower one, and elements of the lower one are forms (representations) of elements of the upper one. Going from lower layers to higher layers corresponds to going from the surface representation to the meaning of text, and vice versa.

The FGD used five different layers:

tectogrammatical layer

surface-syntactic layer

morphological layer

morphonological layer

phonetic layer

The theory of valency, introduced in 3.1, belongs to the tectogrammatical representation of the sentence. Valency is understood as an attribute of autosemantic lexical units. On the tectogrammatical level we assume that every verb, noun, adverb, and adjunct has valency, which is described by valency frames, as already described in 3.1.

Chapters 4.2 and 4.3 describe the data corpora and corresponding valency lexicons that we used in our experiments.

4.2 VALLEX and VALEVAL

4.2.1 VALLEX

VALLEX [Žabokrtský and Lopatková, 2004] is a manually created valency lexicon of Czech verbs, which is based on the framework of Functional Generative Description (see Section 4.1).

The construction of VALLEX started in 2001 and the work is still in progress. The VALLEX version 1.0 [1] (autumn 2003) [Lopatková et al., 2003] which we used in our task and which was published in 2003 defines valency for over 1,400 Czech verbs and contains over 3,800 frames. In 2005, the VALLEX version 1.5 was published, containing roughly 2500 verbs with more than 6000 valency frames. At the time this thesis is submitted, the new version 2.0 of the VALLEX is about to be published. This version of the lexicon uses alternation-based approach [Lopatková et al., 2006]. Alternations are

[1] http://ckl.ms.mff.cuni.cz/zabokrtsky/vallex/1.0/

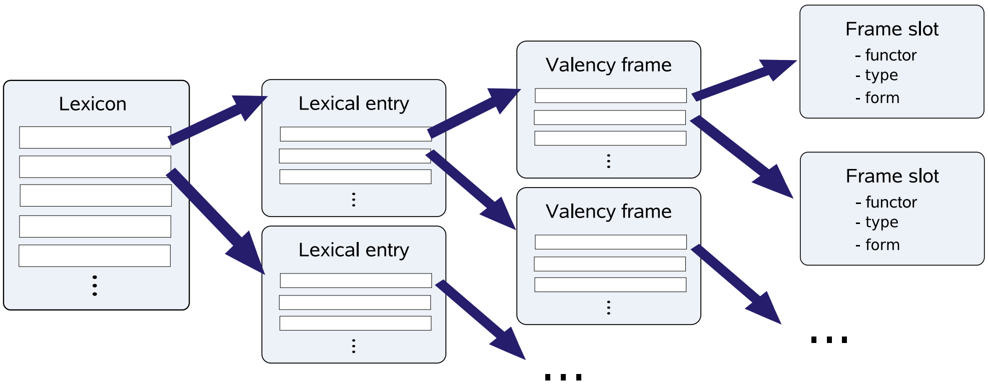


Figure 4.1: Structure of VALLEX and PDT-VALLEX lexicons.

transformation of lexical units describing regular changes in valency structure of verbs.

The basic structure of the VALLEX lexicon is shown in Figure 4.2.1[2]. Elements of the chart are described in the following text in more detail.

The VALLEX lexicon consists of **word entries** corresponding to particular verb lexemes, i.e. complex units consisting of the verb base lemma and possible reflexive particle *se* or *si*. For example, the verb lexeme *dodat si* consists of the base lemma *dodat* and the reflexive particle *si*. There is also the verb *dodat* with no reflexive particle which has another meaning and a different word entry in the lexicon.

Each word entry consists of a definition of one or more **valency frames** which roughly correspond to the senses of the verb. The average number of frames per verb lexeme in VALLEX is 2.7, and the average number of frames per base lemma is 3.9.

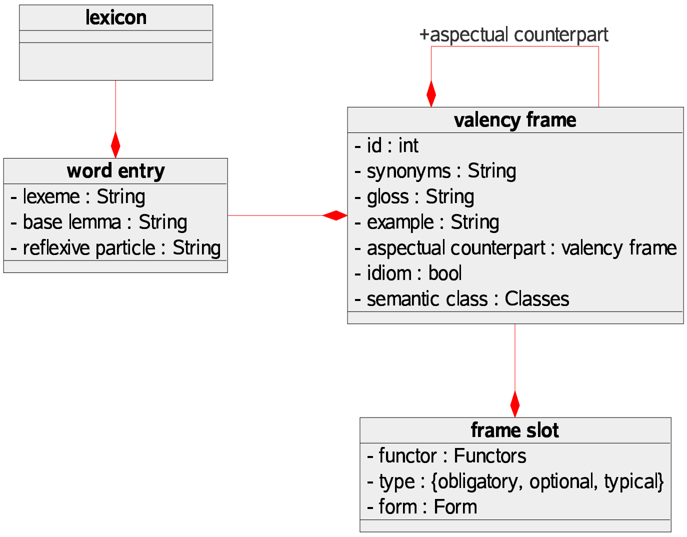


Figure 4.2: The UML class diagram of VALLEX lexicon.

Each valency frame contains a set of **frame slots** corresponding to complements of the verb. Each frame slot is described by a functor, expressing the type of relation between the verb and the complement (e.g. *Actor*, *Patient*, *Addressee*, . . .), a list of the possible morphological forms in which the frame slot might be expressed in a sentence, and the slot type (*obligatory*, *optional* or *typical*).

[2] The rough structure of PDT-VALLEX is the same as that of VALLEX.

Moreover, each frame in the lexicon is accompanied by an explanation of the meaning (using synonyms or glosses), a sample sentence or phrase, and its aspectual counterpart if it exists. Some frames are assigned to semantic classes. A frame could also be marked as “idiom” if it describes an idiomatic usage of the verb.

Figure 4.2 shows the UML class diagram of the structure of the VALLEX lexicon.

Figure 4.3 shows an example of a VALLEX entry for the verb lexeme *dodat*, containing five frames for its different senses, namely *supply*, *ship*, *mention*, *add*, and *encourage*.

4.2.2 VALEVAL

The manually annotated corpus VALEVAL [Bojar et al., 2005] was created in 2005 as a lexical sampling experiment for the VALLEX lexicon. It contains frame annotations for 109 base lemmas selected from VALLEX. As stated in the previous section, the term **base lemma** is used for a lemma excluding its possible reflexive particle.

For the purpose of VALEVAL, the reflexivity of verbs (expressed by a separate reflexive particle) was disregarded, as there is no automatic procedure to determine it. Frames for different lemmas with the same base lemma are all treated as frames of the same lexical unit, and the reflexivity resolution becomes a subtask of the verb sense disambiguation. For example lemmas *brát*, *brát si*, and *brát se* have the same base lemma *brát*, and they all belong to the same part (subtask) of the corpus.

For all verbs in VALEVAL, their aspectual counterparts, including iterative forms, were added too. For each base lemma, 100 sentences from the Czech National Corpus[3] [Kocek et al., 2000] (a large corpus containing over 100 million of words) were randomly selected to be present in VALEVAL.

In order to cover both “easy” and “difficult” cases, verbs were selected randomly from both ends of the difficulty spectrum. Moreover, some verbs were also added on purpose to cover specific cases (e.g. very difficult ones). This selection resulted in an average number of frames per base lemma of 6.77 (according to VALLEX definition).

VALEVAL was concurrently annotated by three annotators looking at the sentence containing the verb and three preceding sentences. Annotators also had the option of selecting no frame if the corresponding frame was missing or if the decision could not be done due to wrong morphological analysis. The inter-annotator agreement of all three annotators was 66.8%,

[3] http://ucnk.ff.cuni.cz/english/index.html

dodat pf.

$\boxed{1}$ dodat$_1$ ≈ **dopravit**
–frame: $\mathbf{ACT}_{1}^{obl}$ $\mathbf{ADDR}_{3}^{obl}$ $\mathbf{PAT}_{4}^{obl}$ $\uparrow\mathbf{DIR}^{typ}$
–example: *dodat někomu zboží do domu*
–asp.counterparts: dodávat$_1$ impf.
–class: transport / exchange

$\boxed{2}$ dodat$_2$ ≈ **dopravit**
–frame: $\mathbf{ACT}_{1}^{obl}$ $\mathbf{PAT}_{4}^{obl}$ $\uparrow\mathbf{DIR3}^{obl}$ $\mathbf{BEN}_{3,pro+4}^{typ}$
–example: *dodat někomu / pro někoho do domu zboží*
–asp.counterparts: dodávat$_2$ impf.
–class: transport

$\boxed{3}$ dodat$_3$ ≈ **říci; podotknout**
–frame: $\mathbf{ACT}_{1}^{obl}$ $\mathbf{PAT}_{k+3}^{opt}$ $\mathbf{EFF}_{4,\check{z}e}^{obl}$
–example: *dodal k tomu své připomínky / vše, co věděl*
–asp.counterparts: dodávat$_3$ impf.
–class: communication

$\boxed{4}$ dodat$_4$ ≈ **doplnit; připojit**
–frame: $\mathbf{ACT}_{1}^{obl}$ $\mathbf{PAT}_{4}^{obl}$ $\mathbf{EFF}_{k+3}^{obl}$
–example: *dodal ke starému zboží nové*
–asp.counterparts: dodávat$_4$ impf.
–class: combining

$\boxed{5}$ dodat$_5$ ≈ **povzbudit** (idiom)
–frame: $\mathbf{ACT}_{1}^{obl}$ $\mathbf{ADDR}_{3}^{obl}$ $\mathbf{PAT}_{2,4}^{obl}$
–example: *dodat někomu odvahy / odvahu*
–asp.counterparts: dodávat$_5$ impf.
–class: exchange

Figure 4.3: Example of VALLEX entry for verb lexeme *dodat* (meanings: *supply*, *ship*, *mention*, *add*, and *encourage*).

the average pairwise match was 74.8%.

4.3 Prague Dependency Treebank

The Prague Dependency Treebank (PDT) [Hajič, 2004] is a manually annotated corpus based on the theory of Functional Generative Description, introduced in Section 4.1. Data of the PDT are part of the Czech National

Corpus [Kocek et al., 2000].

Data are annotated on three different layers [Hajičová, 2002], namely morphological, analytical, and tectogrammatical. This differs from the original definition of layers in the FGD.

Whereas the **morphological layer** deals with individual words, the higher levels (analytical and tectogrammatical layer) use the tree-based (syntactic) sentence structure. The **analytical layer** consists of (surface) syntactic annotation in terms of dependency relations (subject, object, ...). The nodes of the tree are all, as well as the only lexical items from the surface representation of the sentence. The **tectogrammatical layer** describes the underlying syntactic structure – a sentence is described in terms of tectogrammatical dependencies (actor, patient, ...). Abstracting from the surface representation, only auto-semantic words remain in the tree and items from the tectogrammatical structure which are deleted in the surface (dropped pronouns, etc.) shape of the sentence are reconstructed.

The PDT contains yet another layer, **word layer**, which does not contain any annotation, but corresponds to the source data as obtained from the lexical sources (mainly newspapers).

The current version of the Prague Dependency Treebank is version 2.0 which was published by the Linguistic Data Consortium in late 2006 under the number *LDC2006T01*.

Different layers contain different amounts of data. The data are organized so that each part annotated on a higher level is also annotated on all lower levels:

- The morphological layer contains nearly 2 million annotated tokens.
- The analytical layer contains more than 1.5 million tokens.
- The tectogrammatical portion of the corpus contains ca. 800 thousand tokens.

Moreover, the data in each section are divided into the training part, the development testing part (*dtest*), and the evaluation testing part (*etest*). The training part contains approximately 80% of the entire portion, the testing parts each contain approximately 10% of the data.

As frame annotation belongs to the tectogrammatical level, we were restricted to the tectogrammatically annotated portion of the data.

Tectogrammatical annotation in the PDT is done by one annotator only[4], and the consistency was controlled by automatic post-annotation checking [Štěpánek, 2006].

4.3.1 PDT-VALLEX

PDT-VALLEX [Hajič and Honetschläger, 2003], [Hajič et al., 2003] is a valency frames lexicon, created as a part of the PDT. It contains the definition of valency frames for four parts of speech – verbs, nouns, adjectives and adverbs. The PDT-VALLEX was created during the annotation and it contains all auto-semantic words occurring in the corpus. The lexicon was dynamically updated as the annotation went on, unlike VALLEX, described above.

The structure of the lexicon is very similar to the structure of the VALLEX lexicon. The lexicon is composed of **lexical entries**, corresponding to verb lexemes, a base lemma and a possible reflexive particle *se* or *si*. Each verb entry consists of one or more **valency frames** which roughly correspond to different senses of the verb. The average number of frames per verb lexeme in PDT-VALLEX is 1.67, and the average number of frames per base lemma is 1.9. Each valency frame consists of a set of **frame slots** corresponding to complements of the lexical item. Each frame slot is described by a tectogrammatical functor, expressing the type of relation between the main frame element and the complement (this functor must be assigned in the corpus to the tectogrammatical node filling this slot), a description of the possible morphological realization of the slot, and the slot type (*obligatory*, *optional* or *typical*).

Figure 4.4 shows the UML class diagram of PDT-VALLEX lexicon.

Auto-semantic words in the PDT are assigned to valency frames in the PDT-VALLEX. Complements of the auto-semantic words are implicitly mapped to the frame slots, using the tectogrammatical functor.

[4] Each sentence is annotated by one annotator, although there were more annotators altogether.

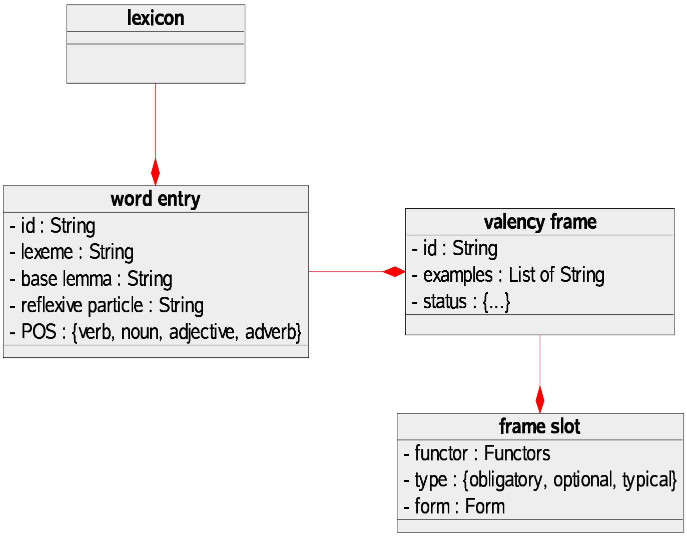


Figure 4.4: The UML class diagram of VALLEX lexicon.

4.4 Comparison of Data Resources

4.4.1 VALLEX vs. PDT-VALLEX

The following list summarizes the main differences between the two valency lexicons introduced in this chapter, VALLEX and PDT-VALLEX:

- VALLEX was created from scratch, without any explicit demand for data annotation (lexicon first), while PDT-VALLEX was primarily created for the annotation of the PDT (corpus first).

- The lemma selection in VALLEX was led by linguistic intuition, common but uninteresting verbs might be missing.
  The lemma selection in PDT-VALLEX was led by occurence in the PDT, it captures all verbs (and frames) occurring in the corpus.

- In VALLEX, complete lemma records, consisting of the list of frames, were included at once, while in PDT-VALLEX, frames were included for corresponding lemmas sequently, as they were annotated in the corpus.

The following table compares the basic quantitative characteristics of the lexicons:

	VALLEX	PDT-VALLEX
Number of verb entries	2,476	5,510
Average frames per verb	2.63	1.67

4.4.2 VALEVAL vs. PDT

The following list summarizes the main differences between the two corpora introduced in this chapter – the VALEVAL and the tectogrammatical part of the Prague Dependency Treebank (PDT):

- VALEVAL only contains the valency annotations of verbs, while PDT is a complex corpus with the annotation of all words.
- VALEVAL was created to serve an existing lexicon, while PDT was created from scratch as the primarily created data resource.
- VALEVAL contains only one annotation for each sentence. If one sentence contains several verbs from the set of VALEVAL verbs, it might be present more times with different annotated verbs. In PDT, each sentence contains the annotation of all occurred verbs (nouns, adjectives, and pronouns).
- VALEVAL contains sentences randomly picked from the Czech National Corpus, while PDT contains whole documents.
- VALLEX contains the annotation of 109 different verbs, while PDT contains the annotation of 4,845 different verbs.
- Valency in VALEVAL were annotated by three annotators in parallel, so that inter-annotator agreement could have been computed and a corpus of golden annotation could have been prepared. In PDT, the valency of each word were annotated by a single annotator only.

Chapter 5

Related Work

In this chapter we summarize related works concerning the verb sense disambiguation. We mention different Czech and English lexicons of verbs (possibly with their corresponding corpora) which we did not use in this work but which are comparable to the ones we did. Our methods might be with more or less modifications applied to those sources, too. We also introduce SensEval, a multilingual word sense disambiguation competition, and its successive, SemEval. Different approaches to word sense disambiguation are also mentioned.

5.1 Verb Senses Lexicon

5.1.1 BRIEF

BRIEF [Karel Pala, Pavel Ševeček, 1997] is an electronic valency lexicon for Czech containing roughly 15,000 verbs.

Each verb is described by a list of one or more valency frames. A valency frame consists of frame elements determined by their surface form.

The frame determination in the lexicon is driven merely by the surface syntax. This means that a given sense, possibly realized by two syntactic (surface) forms is represented by two different frames, and there is no relation joining them. On the other hand, if two different senses are realized equally on the surface, they share the same frame and there is no way to distinguish them in the means of the lexicon. This property better reflect the primary definition of the valency frames[1], on the other hand, this makes it less suitable for using for word sense disambiguation.

Compared to VALLEX and PDT-VALLEX, the BRIEF lexicon does not describe semantic functions, only the forms are described. However, the frame elements might be assigned semantic features for person or thing.

[1] Compared VALLEX and PDT-VALLEX

Czech syntactic lexicon [Hana Skoumalová, 2001] is an extension of the BRIEF lexicon, adding tectogrammatical functors, obligatorness, reflexivity, subject specification, control and other linguistical information. The syntactically driven frame distinction remains.

5.1.2 WordNet

WordNet [Fellbaum, 1998] is an English lexicon created at the Cognitive Science Laboratory at the Princeton University. Currently, WordNet is recognized as one of the most important lexical resources in computational linguistics.

The current version, WordNet 3.0, contains over 150 thousands of lexical entries for open-class words – nouns, verbs, adjectives and adverbs. The most attention is given to the nouns.

The lexical items are organized into semantic units, so called **synsets** (sets of cognitive synonyms), which correspond to senses of words. Homonymous words belong to more different synsets, as well as more (synonymous) words can belong to the same synset. WordNet organizes synsets into semantic relations (hyponyms/hyperonyms, meronyms, ...), creating a semantic map of all synsets. The current version contains over 117 thousands of synsets.

WordNet was used as the sense inventory in the English task of Senseval-2 and Senseval-3, see Section 5.2.1. The original English WordNet has also been converted to several other languages, resulting in many current projects, like Chinese WordNet, or EuroWordNet (see Section 5.1.3).

The WordNet is freely available for research purposes and also an on-line version is accessible.[2]

5.1.3 EuroWordNet

EuroWordNet [Vossen et al., 1998] is a multilingual lexical database for seven European languages: Dutch, Italian, Spanish, German, French, Czech ([Pala and Smrž, 2004]) and Estonian. The project started form the English lexicon WordNet and it has the same data structure and format.

The EuroWordNet is not a straight-forward translation of the WordNet into the target languages, as the word senses often do not correspond one-to-one across different languages.

The mapping between languages is provided by Inter-lingual index, a system of numerical identifiers which are unique across all the languages.

[2] `http://wordnet.princeton.edu/perl/webwn?s=word-you-want`

5.1.4 FrameNet

FrameNet [Baker and Sato, 2003] is a project created at the Berkeley University that creates a large semantic lexicon of English for NLP applications providing information on predicate-argument structure. FrameNet is based on the theory of frame semantics, originally introduced by Fillmore in ([Fillmore, 1976]).

Frames are considered to be conceptual structures or prototypical situations. They are evoked by predicates (**frame evoking elements, FEE**s) and they are associated with other constituents (**frame elements, FE**s) which correspond to the participants of the situations.

A particular combination of frame elements in FrameNet is local to a given frame – their names are domain specific [Johnson and Fillmore, 2000] (e.g. Speaker, Message and Topic in Communication frame) – some of the frame elements are more general, some of them are specific to a small group of lexical items. A frame definition in the FrameNet database consists of a frame description and a list of frame elements and their descriptions. Moreover, the frame definition is also accompanied by a list of predicates (verbs and nouns) that can evoke this frame, i.e. can serve as frame evoking elements of a particular frame (e.g. frame Communication can be evoked by the verbs *speak*, *talk*, the noun *dialog*, etc.). Furthermore, FrameNet contains links to other lexical resources – e.g. WordNet. Figure 5.1 presents an example of a FrameNet frame definition.

Sentences are described in terms of frames, each frame is evoked by one frame evoking element and some of its frame elements[3] are assigned to syntactic constituents of the sentence. Figure 5.2 shows an example sentence with an assigned Statement frame.

FrameNet defines relation of inheritance among frames, a frame can inherit from one or more other frames. For example, Statement and Communication_noise inherit from Communication frame. Moreover, FrameNet defines relation of *using*, which describes using of a frame within another frame, e.g. Communication frame uses Topic frame and is used by Attempt_ suasion, Candidness, Commitment, and other frames.

The FrameNet database is accessible via Internet at the address of the FrameNet project[4] and currently contains 482 frames and thousands of lexical entries.

[3] Not all frame elements have to be present in the sentence (i.e. event).

[4] http://www.icsi.berkeley.edu/∼framenet/

Frame:	STATEMENT	This frame contains verbs and nouns that communicate the act of a Speaker to address a Message to some Addressee using language. A number of the words can be used performatively, such as *declare* and *insist*.
Frame elements:	*Speaker*	is the person who produces the Message (whether spoken or written). It is normally expressed as the External Argument of predicative uses of the TARGET word, or as the Genitive modifier of the noun.
	Addressee	receives a Message from the Communicator (Speaker).
	Message	is the FE that identifies the content of what the Speaker is communicating to the Addressee. It can be expressed as a clause or as a noun phrase.
	Medium	is the physical entity or channel used by the Speaker to transmit the statement.
	Topic	The Topic is the subject matter to which the Message pertains. It is normally expressed as a PP Complement headed by "about", but in some cases it can appear as a direct object.
Frame evoking elements:	*add.v, address.v, admission.n, admit.v, affirm.v, affirmation.n, allegation.n, allege.v, announce.v, announcement.n, assert.v, assertion.n, attest.v, aver.v, avow.v, avowal.n, boast.n, boast.v, brag.v, caution.v, claim.n, claim.v, comment.n, comment.v, complain.v, complaint.n, concede.v, concession.n, confess.v, confession.n, ...*	

Figure 5.1: Example of STATEMENT frame definition

[Frank and Semecký, 2004] describes a method for corpus-based induction of an LFG syntax-semantics interface for frame semantic processing in a computational LFG parsing architecture using FrameNet as the semantic lexicon.

Speaker	FEE	Addressee	Medium
Kim	*QUESTIONED*	*me*	*over the phone.*

Figure 5.2: Sentence with assigned Statement frame

5.1.5 Proposition Bank

Proposition Bank (PropBank) [Kingsbury et al., 2002] is a project of the University of Pennsylvania which aims at adding a layer of semantic annotation to the Penn English Treebank [Marcus et al., 1994].[5] The basis for semantic annotation are syntactically hand-annotated sentences from the Penn Treebank II Wall Street Journal corpus of a million of words.

Each predicate defined in PropBank is assigned arguments which are numbered sequentially as **Arg0**, **Arg1**, **Arg2**, . . . , and the numbering is predicate dependent. **Arg0** is usually the subject of a verb, **Arg1** direct object of a transitive verb, etc. This is a conceptual difference from the FrameNet project, in which semantic roles are given meaningful frame dependent names, i.e. predicates of the same frame share the role names. Arguments in PropBank are, nevertheless, given mnemonic labels too. These labels are verb specific, however some of them tend to be specific to a group of verbs, closer to FrameNet conventions.

In addition to numbered arguments, a predicate can be assigned additional mandatory adjuncts[6], which are not numbered but rather labeled with ‘ArgM-’ extended with a secondary functional tags: (LOC for location, TMP for time, MNR for manner, DIR for direction, CAU for cause, NEG for negation marker, MOD for modal verb, PRP for purpose, and ADV for general-purpose modifier). Secondary predication is marked with tag PRD in the cases where one argument of a verb is a predicate upon another argument of the same verb.

In PropBank, verbs take usually three or four arguments:

obtain.01 (“get”)	
Arg0:	receiver
Arg1:	thing gotten
Arg2:	received from

[5] http://www.cis.upenn.edu/∼treebank/

[6] If the predicate requires the particular adjunct strongly enough.

They can take no arguments (e.g. weather predicates):

hail.01 (“weather phenomenon”)

Maximally, some verbs take six arguments:

edge.01 (“move slightly”)	
Arg1:	Logical subject, patient, thing moving
Arg2:	EXT, amount moved
Arg3:	start point
Arg4:	end point
ArgM-LOC:	medium
Arg5:	direction–REQUIRED

The semantics of arguments is predicate dependent but it follows certain guidelines. The authors try to keep consistency across semantically related verbs. For instance *buy* and *purchase* have the same set of arguments, and they are similar to the set of arguments of *sell*, cf. Figure 5.3. However, two senses of a single verb can have different argument labels.

Figure 5.4 shows an example of PropBank annotation.

Purchase	**Buy**	**Sell**
Arg0: buyer Arg1: thing bought Arg2: seller Arg3: price paid Arg4: benefactive	Arg0: buyer Arg1: thing bought Arg2: seller Arg3: price paid Arg4: benefactive	Arg0: seller Arg1: thing sold Arg2: buyer Arg3: price paid Arg4: benefactive

Figure 5.3: Semantic roles of predicates *buy*, *purchase*, and *sell*

Arg0	REL	Arg1	Arg3
The holder	*buys*	*$1000 principal amount*	*of debentures at par.*

Arg0	REL	Arg4	Arg1
John	*bought*	*his mother*	*a dozen roses.*

Figure 5.4: Sentences with PropBank annotation

5.2 Word Sense Disambiguating

> "One of the most significant problems in processing natural language is the problem of ambiguity. Most ambiguities escape our notice because we are very good at resolving them using context and our knowledge of the world. But computer systems do not have this knowledge, and consequently do not do a good job of making use of the context."
>
> *Cecilia Quiroga-Clare*
> Language Ambiguity: A Curse and a Blessing.
> [Quiroga-Clare, 2003]

The ambiguity is an ubiquitous property of language, present at different levels – phonetic ambiguity, morphological ambiguity, lexical ambiguity, structural ambiguity, semantic ambiguity. In this work we concentrate solely on the lexical ambiguity or, to be more precise, lexical disambiguation (resolving the ambiguity).

There is a recently published book [Agirre and Edmonds, 2006] dedicated to different aspects of word sense disambiguation. It is a overview monography, introducing different aspects of the topic – it mentions several supervised and unsupervised methods, discusses available lexical resources, as well as deals with the problem of sense definition.

5.2.1 Senseval

In relation to lexical word sense disambiguation, we can not forget to mention Senseval[7], an evaluation exercises for the semantic analysis of text. Senseval is a public contest designed to compare different WSD solutions and attempts. Senseval contains different tasks for different languages, the preparation phase usually takes about 6 month in which participants prepare their solutions of the tasks and the presentation phase which has a form of workshop. Senseval already had three runs. **Senseval-1** (1998) included task for English, French, and Italian. **Senseval-2** (2001) included tasks for Basque, Chinese, Czech, Danish, Dutch, English, Estonian, Italian, Japanese, Korean, Spanish, Swedish. **Senseval-3** (2004) included tasks for English, Italian, Basque, Catalan, Chinese, Rumanian, and Spanish, as well as tasks dedicated to concrete applications (MT task, semantic roles disambiguation task, identification of logic forms).

[7] `http://www.senseval.org/`

Semeval-1 / Senseval-4 is currently underway, it will take place by the ACL 2007 in Prague.

There is a large number of prior research papers on word sense disambiguation.

As there are dozens of resources dedicated to the problem of word sense disambiguation, we try to pick out those which deal with similar aspects as this work does – those, which compare different supervised methods, or different approaches to describe word (preferably verb) occurences, or those which analyze Czech language.

[**Dang and Palmer, 2005**] describes statistical system for disambiguation of verb senses using different features. According to authors, the system performed at best published accuracy on the English verbs of Senseval-2. The authors divide features into groups comparable to our approach:

- **Topical features** describe occurences of keywords (anywhere) in the sentence. Those features correspond to syntactically non-bounded WordNet features and idiomatic features in our experiments.
- **Collocational features** describe lemmas and tags in the neighborhood of the disambiguated verb. Those features roughly correspond to morphological features used in our experiments. However, because of the rich Czech morphology, our morphological features covers more information.
- **Syntactic features** are boolean features following from the syntactical tree of the sentence. Those features correspond to syntax-based features in our experiments. Again, because of the complexity of the language, we use richer feature set.
- **Semantic features** describe semantic class information. Those features roughly correspond to WordNet, and animacy features used in our experiments.

The authors showed that adding features from richer linguistic sources always improves accuracy. However, they claim that the topical features did not improve the accuracy significantly, because the most of the information provided by the topical features were already captured by the features from the richer linguistic sources.

Further, the authors added features using manually annotated PropBank roles and labels (only gold-standard data were used). The accuracy rises significantly, however this is a questionable comparison, as the manually annotated data were used. Moreover automatic semantic role labeling is claimed to be a difficult task, and the predicate-argument PropBank annotation already contain a lot of semantic information and sense distinction.

The system used Maximum Entropy model, whereas our system compares different models (including Maximum Entropy).

[**Escudero et al., 2000**] compares different classification algorithms on DSO corpus – Naïve Bayes, Exemplar-based (using k-nearest neighbor), Winnow-based, and LazyBoosting (an adaptation of AdaBoost) algorithms.

[**Lee and Ng, 2002**] explores the relative contribution of different knowledge sources and learning algorithms to WSD for different part of speech.

They used following types of features:

- Part of speech of neighboring words, similar to our morphological features. However, because of the rich Czech morphology, our feature set is much more complex.
- Words in surrounding context, roughly correspond to our syntacticaly non-bounded WordNet and idiomatic features.
- Local collocation of words in neighbor of the disambiguated verb.
- Syntactic relations, corresponding to our syntax-based features. Similarly to our work, they used Charniak's parser and dependency trees.

Features captured part of speech of neighboring words, words in surrounding context, local collocation, and syntactic relations.

Similar to our work, this paper compares different classification algorithms, including Support Vector Machines, Naïve Bayes Classifiers (NBC), AdaBoost, and decision trees. In our experiments, we did not use AdaBoost approach, but on contrary we used Maximum Entropy model. Authors showed that prior reduction of the feature space helped some algorithms (NBC, decision trees), while hurt others (SVM, AdaBoost). In our work, we used the feature space reduction for decision trees and we also noticed a significant improvement.

The authors showed that differences in results of different feature types depended on the chosen method.

[Florian et al., 2002] used similar features as the approach of [Dang and Palmer, 2005]. Their features were based on information about raw words, lemmas, part of speech tags, and syntactic relations. The authors showed different classifiers combination methods, what further improved their results (by 1.3% on English).

First results of verb frames disambiguation has already been reported. [Erk, 2005] describes a frame assignment task as a special type of word sense disambiguation and gives limited results for German.

[Lopatková et al., 2005] and [Semecký, 2006] gives results of the experiment of disambiguation of Czech valency frames on the VALEVAL corpus. These were the results from the initial phase of the experiment thoroughly described in this thesis.

First results of the disambiguation of valency frames on the Prague Dependency Treebank were reported in [Semecký and Podveský, 2006].

[Král, 2001] show an experiment of WSD for Czech on a small corpus containing five polysemous nouns. [Král, 2002] and [Král, 2004] presents different approaches to the WSD – an approch using morphological characteristics of the disambiguated word, a bag-of-words approach, and an approach using clustering contexts of words.

[Cikhart and Hajič, 1999] presents WSD of Czech juridical texts. The authors use combination of Naïve Bayes, Decision Lists and hand-written rules, and evaluate the method with regard to the information retrieval task.

[Rivest, 1987] and [Yarowsky, 1994] use Decision Lists for word sense disambiguation, too.

Chapter 6

Feature Design

The following two chapters are the main contribution of the thesis. We introduce the design of features proposed for this task, thoroughly describe experiments of verb sense disambiguation which we performed and we give a detailed evaluation. Different aspects of the disambiguation are task and evaluated separately.

The basic design of the experiments consists of the following steps:

1. **Preparation of data.** Data from different sources (corpora) are prepared for the process. This basically means converting data from different sources to the same format. The preparation of data is described in section Section 6.1.

2. **Feature generation.** Data, which are to be disambiguated, are sentences described by complex linguistic structures. In order to involve them in the learning and testing process, we convert them into vectors of features. Different types of features are proposed, generated and separately evaluated. We also try using various combinations of feature types. The generation of features is described in this chapter.

3. **Machine learning.** Later, feature vectors are used to train classifiers – algorithms for automatic labeling of data. Different classifiers are suitable for different applications. In our experiments, we tried several types of classifiers to find out which one fits best which type of task and features. The application of the methods is described in Section 7.1.

   As the sets of the verb senses (or the verb frames) differ for each verb, the disambiguation task consists of several independent subtasks, one for each verb (base lemma). Each subtask can be trained only on a corresponding subset of training data which increases the need of large training data.

4. **Evaluation.** Finally, evaluation is performed. We evaluated diverse combinations of data, types of features and classifiers. Moreover, we

give different types of evaluation with regard to the types of data. Evaluations are described in detail in Chapter 7.

6.1 Data Preparation

VALEVAL

The VALEVAL corpus was described in Section 4.2.2. It consists of 100 sentences for each of the 109 verbs selected from the Czech National Corpus. The VALEVAL corpus was originally created in an XML format as a single file containing the whole corpus.

For input data for the frame disambiguation task, we used VALEVAL sentences where all three annotators agreed. Moreover, sentences on which annotators did not agree were rechecked by another annotator, and sentences with a clear mistake were corrected and added too. This resulted in a set of 8,066 sentences.

For better manipulation with the data, we translated the format into several files, one for each base lemma. The corpus contained only raw texts. However, in the disambiguation procedure, we needed linguistically analyzed data, therefore we processed morphological tagging and surface syntactical parsing first.

Tagging. We used a morphological tagger created by Jan Hajič [Hajič, 2000]. This tagger was trained on Prague Dependency Treebank 1.0.

Parsing. We used the Charniak's parser [Charniak, 2000] trained on the Prague Dependency Treebank 1.0. The corpus contains among others fictional texts and some sentences come from text sources which do not use sentence punctuation, the author intentionally continues without finishing sentences. Therefore some sentences reached a length of a couple of hundred words. This caused the parser difficulties and some sentences could not be parsed at all.

After excluding unparsed sentences, 7,778 sentences remained which served as input for disambiguation methods. There were 61.2 sentences per base lemma on average, ranging from a single sentence to 100 sentences (the original amount in the VALEVAL).

Figure 6.1 shows the distribution of the number of sentences per base lemma.

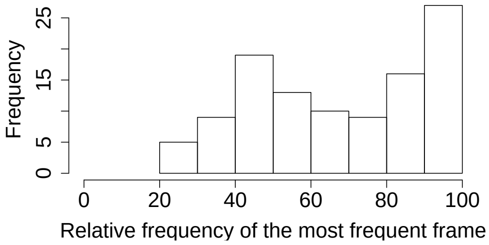


Figure 6.1: Distribution of the number of sentences per base lemma in VALEVAL

Prague Dependency Treebank

For the purpose of our task, we considered only verbs that appeared at least once in both the training set and the testing set. Again, the reflexivity was disregarded.

The sentences in PDT were automatically analytically parsed by MST parser [McDonald et al., 2005] using the Jack-knife method.

There were 46.03 sentences per base lemma on average, ranging from two to 11,345 sentences for the verb "být" (*to be*). The number of sentences for each verb in PDT reflects the real data distribution. Figure 6.2 shows the distribution of the number of sentences per base lemma in PDT.

Compared to VALEVAL, PDT contains many more sentences and the distribution sample count reflects the distribution in real data, following the Zipf's law. VALEVAL, on the other hand, is more reliable, because only the golden annotations were used in our task.

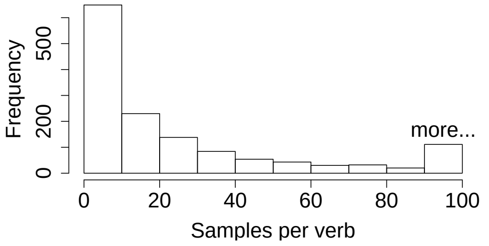


Figure 6.2: Distribution of the number of sentences per base lemma in PDT

To disambiguate a word or a phrase, we are looking at linguistics characteristics within its context. The extend of the context depends on the method and on the type of disambiguated object. It can be surrounding words, a whole sentence, paragraph or document. In our work, we look at the sentence in which the verb occurs.

The linguistic characteristics of a sentence are complex structures – trees, vectors, sets, ... On the contrary, machine learning methods can only deal with a simple description of samples, usually vectors.

The natural solution to deal with this contrast is to convert complex linguistic characteristics into simple vectors of features. As the vectors of features only describe linguistic information in a limited way, there will always be a loss of information in the feature creation process. Therefore the selection of a suitable set of features is essential for the success of the method.

In this section, we thoroughly inscribe types of features which we use for the disambiguation task, we describe why we used each type and how we generated them from linguistic data. We experimented with five types of features, all of them describe different information about the context of the verb within one sentence.

The types of features are:

- **Morphological**: purely morphological information about lemmas in a small window centered around the verb.
- **Syntax-based**: information resulting from the output of the automatic syntactic parser (including mainly morphological and lexical characteristics).
- **Idiomatic**: occurrence of idiomatic expressions in the sentence according to the VALLEX lexicon both dependent on the verb and occurring anywhere in the sentence.
- **Animacy**: information about the animacy of nouns and pronouns both dependent on the verb and occurring anywhere in the sentence.
- **WordNet**: information based on the WordNet top-ontology classes of the lemmas both dependent on the verb and occurring anywhere in the sentence.

Table 7.10 shows the number of features belonging to each of the groups. In the following sections, we give a detailed description of each group of features. All the features are gained from the data preprocessed by statistical methods only, i.e. no human annotation is used. The complete list of features is given in Appendix A.

Types of features	**#Features**
Morphological	60
Syntax-based	103
Idiomatic	118
Animacy	14
WordNet	128
Total	**423**

Table 6.1: Types of features.

6.2 Morphological features

These features are generated only from the morphological information, they do not use parsing.

Because syntactical parsing is computationally much more demanding than morphological tagging, those features are very simple and easy to obtain.

The morphological features are based on the Czech positional morphology ([Hajič, 2000]) used in the Prague Dependency Treebank. The morphological tags consist of 15 positions (characters), each stating the value of one morphological category, see Table 6.2.

In this work, we use all positions of the morphological tags, except positions 13, 14, and 15, which are not actively used.

Categories which are not relevant for a given lemma (e.g. tense for nouns) are assigned a special value (“–”).

6.2.1 Feature description

For lemmas within a n-word window centered around the verb we used each position as a single feature. Originally, we used a five-word window (two preceding lemmas, the verb itself, and two following lemmas), but in later stages, we also experimented with other widths of the window. In Chapter 7, we evaluate the impact of the window width on the performance of the disambiguation.

For the five-words windows, we obtained 60 morphological features – 5 words, 12 features for each.

Implementational note: Morphological category *Detailed part of speech* could contain non-alphanumeric characters which might have a special meaning in subsequent data processing (“#”, “%”, “*”, “,”, “}”, “:”, “=”, “?”,

Position	Name	Description
1	POS	Part of speech
2	SubPOS	Detailed part of speech
3	Gender	Gender
4	Number	Number
5	Case	Case
6	PossGender	Possessor's gender
7	PossNumber	Possessor's number
8	Person	Person
9	Tense	Tense
10	Grade	Degree of comparison
11	Negation	Negation
12	Voice	Voice
13	Reserve1	Reserve
14	Reserve2	Reserve
15	Var	Variant, style

Table 6.2: Categories of the Czech positional morphology.

"@", "ˆ"), therefore we replaced these values with codes:

#	__hash__	%	__percentage__	*	__star__
,	__comma__	}	__rbrace__	:	__doubledot__
=	__equal__	?	__questionmark__	@	__at__
ˆ	__caret__				

Figure 6.3 shows an example of generation of morphological features for verb *odvolat* (call away).

Radní	**také**	**odvolali**	**ředitele**
`AAMP1----1A----`	`Db-------------`	`VpMP---XR-AA---`	`NNMS4-----A----`
Councillors	*also*	*called away*	*the director*
této	instituce	.	
`PDFS2----------`	`NNFS2-----A----`	`Z:-------------`	
of this	*institution*	.	

Figure 6.3: Generation of morphological features.

6.3 Syntax-based features

Syntax-based features, in contrast to the morphological features, are based on the result of the syntactical (analytical dependency) parser. Different corpora were parsed with different parsers, as described in Section 6.1.

Syntax-based features also use morphological characteristics, but combine them with the shape of the dependency tree. As the term *syntactic features* might suggest using only syntactic information by analogy with the *morphological features* using only information about morphology, we prefer to use the term *syntax-based features*. Moreover, other types of features (idiomatic, WordNet-based, and animacy) also use the analytical syntax, however, they are in special categories because of their narrow scope.

For our experiments, we did not use a tectogrammatical parser, as we understand verb valency as a part of the tectogrammatical analysis. Therefore the tectogrammatical parsing and subsequent analysis (assignment of tectogrammatical functions) should be processed only after the valency is resolved.

We expected that syntax-based features would be very useful for the disambiguation of the valency frames as the valency frames describe the syntactical behavior of the verbs. Special care was given to selecting the proper features. Nevertheless, because statistical parsing achieves much lower accuracy than morphological tagging, syntax-based features as opposed to morphological features can suffer much more from errors in analysis.

Based on the results of statistical syntactic parsers we extracted the following groups of features:

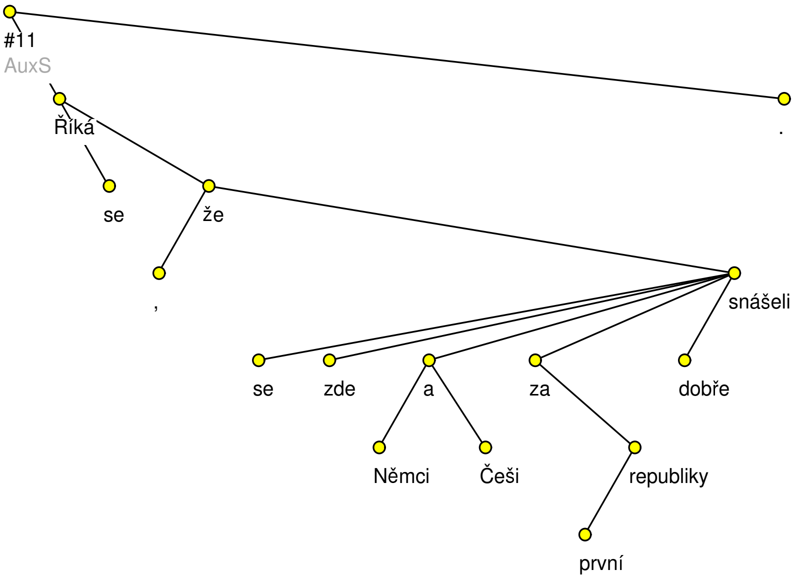
- Reflexive *se*
- Reflexive *si*
- Subordinate verb
- Superordinated verb
- Subordinating conjunctions
- Substantive cases
- Adjective cases
- Prepositional cases

A detailed description of each group follows.

6.3.1 Reflexive *se*.

Boolean feature stating whether there is a reflexive pronoun *se* dependent on the verb. We distinguish between the preposition “se” (morphological tag `RV--7----------`) and the pronoun “se” (morphological tag `P7-X4----------`), and only use the pronouns. On the other hand, we did not distinguish between the reflexive tantum and the optional reflexive pronoun, both denoted by the morphological tag `P7-X4----------`.

Figure 6.4 shows an example of an analytical tree with the base lemmas *říkat* and *snášet*, both with the reflexive particle *se*. The feature will be set to *true* for both verbs.



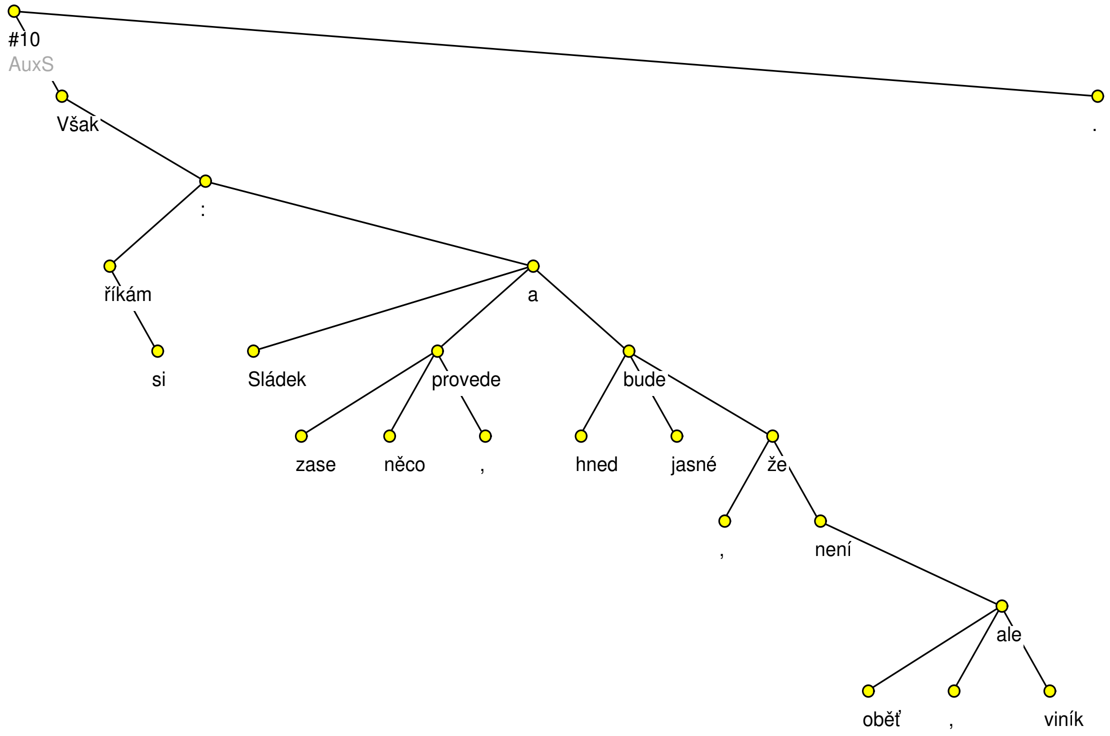
***Říká se**, že **se** zde Němci a Češi za první republiky dobře **snášeli**.*
***It is said** that Czechs and Germans **got along well** here in the First Republic era.*

Figure 6.4: Verb with the reflexive particle *se*

6.3.2 Reflexive *si*.

Boolean feature stating whether there is a pronoun *si* dependent on the verb. Again, we do not distinguish between reflexive tantum and optional reflexive pronoun, both denoted by the morphological tag `P7-X4----------`.

Figure 6.5 shows an example of an analytical tree with the verb *říkat* with the reflexive particle *si*. For this verb the feature will be set to *true*. For the verbs *provést* and *být*, also contained in the sentence, the feature will be set to *false*.



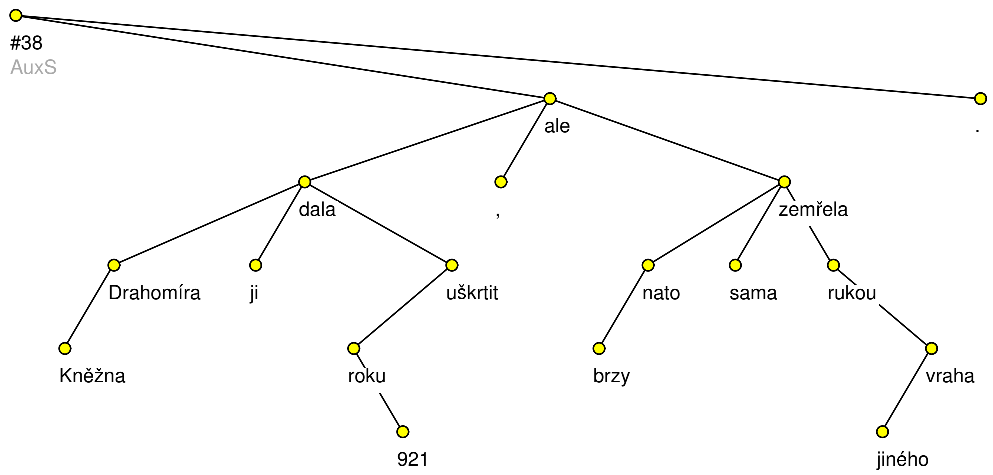
Však ***říkám si****: Sládek zase něco provede, a hned bude jasné, že není oběť ale viník.*
But I ***say to myself****: Sládek will do sometimes again, and ...*

Figure 6.5: Verb with the reflexive particle *si*

6.3.3 Subordinate verb.

Boolean feature stating whether the analyzed verb depends on another verb.

Figure 6.6 shows an example of a verb *uškrtit* which is a subordinate verb to the verb (ordinated by the verb *dát*, form *dala*).



Princess Drahomíra let her choke to death in 921 but shortly after that she was killed too by another assassin.
Kněžna Drahomíra ji dala roku 921 uškrtit, ale brzy nato sama zemřela rukou jiného vraha.

Figure 6.6: Example of a subordinate and a superordinate verb

6.3.4 Superordinated verb.

Boolean feature stating whether the analyzed verb is a superordinate verb of another verb.

Figure 6.6 shows an example of the verb *dát* (form *dala*) which is a superordinate verb of the verb *uškrtit*.

The features subordinate verb and superordinate verb seem to be important indicators for sense disambiguation, however, they also suffer from errors in statistical parsing because the determination of parent of a (non-root) verb is a hard task (sometimes even for people).

6.3.5 Subordinating conjunctions.

Thirty eight boolean features, one for each subordinating conjunction stating whether a particular conjunction is present among the nodes syntactically dependent on the analyzed verb.

The following table lists the subordinating conjunctions considered in this analysis:

ač	*aniž*	*jako*	*jestli*	*liž*	*přestože*	*seč*	*zda*
ačkoli	*byť*	*jakoby*	*jestliže*	*že*	*přičemž*	*takže*	*zdali*
aby	*co*	*jakož*	*kdyby*	*než*	*pokud*	*třebaže*	
ať	*coby*	*jakožto*	*když*	*nežli*	*poněvadž*	*zatímco*	
až	*jakmile*	*jelikož*	*leč*	*pakliže*	*protože*	*zato*	

6.3.6 Substantive case.

Seven boolean features, one for each case, stating whether there is a noun or a substantive pronoun in the given case dependent on the analyzed verb. Substantives with prepositions are not counted for those features, but they have their own category.

The presence of a noun or a syntactic pronoun is determined by the morphological category of nodes dependent on the verb. Words with the morphological tags matching the following regular expressions are considered as substantive-like and imply setting the appropriate feature to *true*:

`^N`	Nouns
`^P[567DHJKPQ]`	Substantive pronouns
`^Cy`	Numerals, fraction ending at *-ina* (*pětina*, lit. *one fifth*)

Moreover, for some types of pronouns, also the lemma is considered:

`^PW`	Negative pronouns – considered as substantive-like if the lemma is *nic*, *nikdo*, or *pranic*.
`^PZ`	Indefinite pronouns, – considered as substantive-like if the lemma is *něco*, *někdo*, *bůhvíco*, *cokoliv*, *cosi*, *kdeco*, *kdekdo*, *kdokoliv*, *kdosi*, *lecco*, *leccos*, *ledacos*, *leckdo*, *máloco*, *málokdo*, *něco*, *někdo*, or *všelicos*.

The case is taken from the fifth position of the morphological tag.

6.3.7 Adjective case.

Seven boolean features, one for each case, stating whether there is an adjective or an adjective pronoun in the given case directly dependent on the analyzed verb. Substantives with prepositions are not counted for those features, but they have their own category.

The presence of an adjective or an adjective pronoun is determined by the morphological category of nodes dependent on the verb. Words with the part of speech set to `A` (adjectives) imply setting the appropriate feature to *true*. Moreover, for some types of pronouns, also the lemma is considered:

`^A`	Adjectives
`^PW`	negative pronouns – considered as adjective-like if the lemma is *nijaký*, *pražádný*, *žadný*, or *nižádný*.
`^PZ`	indefinite pronouns, – considered as adjective-like if the lemma is *čísi*, *jaký*, *jakýkoliv*, *jakýs*, *jakýsi*, *kterýkoliv*, *kterýsi*, or *takýs*.

The case is taken from the fifth position of the morphological tag.

6.3.8 Prepositional case.

Seven boolean features, one for each case, stating whether there is a prepositional phrase in this case dependent on the verb. The presence of a preposition is determined from the morphological tag *Part of speech* set to *R* – preposition). Again, the case is taken from the fifths position of the morphological tag.

`^R`	Preposition

In our opinion, those features will probably not work very well, as different prepositions with the same case do not suggest related meanings any more than two different prepositions with a different case. However, it is possible that in special cases (where there is little training data), they might work well as they are more coarse-grained compared to the lexicalized prepositional features, whose description follows.

6.3.9 Lexicalized prepositional phrase.

Sixty nine boolean features, one for each possible combination of a preposition and a case, stating whether there is the given preposition in the given case directly dependent on the verb.

If a preposition can be present in two (or more) cases, we introduce a separate feature for each case as the semantics of the cases differs. This concerns prepositions: *mimo* (genitive and accusative), *nad* (accusative and instrumental), *na* (accusative and local), *o* (accusative and local), *pod* (accusative and instrumental), *s* (genitive and instrumental), *v* (accusative and local), and *za* (genitive, accusative, and instrumental).

A list of the lexicalized prepositional features follows (numbers following prepositions signify cases, 1–nominative, 2–genitive, 3–dative, 4–accusative, 5–vocative, 6–local, 7–instrumental):

během-2
bez-2
blízko-2
díky-3
dík-3
dle-2
do-2
kolem-2
kontra-1
kromě-2
k-3
kvůli-3
mezi-7
mezi-4
mimo-4
mimo-2
místo-2
nad-7
nad-4
namísto-2
napospas-3
naproti-3
na-4
na-6
ob-4
od-2
ohledně-2
okolo-2
oproti-3
o-4
o-6
poblíž-2
podél-2
podle-2
pod-4
pod-7
pomocí-2
po-4
prostřednictvím-2
proti-3
pro-4
před-4
před-7
přes-4
při-6
skrz-4
stran-2
s-2
s-7
uprostřed-2
u-2
uvnitř-2
včetně-2
vedle-2
versus-1
vně-2
vstříc-3
v-4
v-6
vůči-3
vůkol-2
vyjma-2
vzdor-3
za-2
za-4
za-7
zpod-2
zpoza-2
z-2

The verb *snášet* (form *snášeli*) in Figure 6.4 has the value of the feature **za-2** set to *true*, the other lexicalized prepositional phrase features are set to *false*.

Altogether we used 103 syntax-based features.

6.4 Idiomatic features

Certain idiomatic expressions evoke a special (usually figurative) senses of verbs. To depict such senses, we introduced this type of features.

Each idiomatic construction (multi-word expression) described in the VALLEX lexicon was used as one boolean feature. This feature was set to *true* if this construction occurred in the raw text of the sentence containing the verb continuously. Features corresponding to not occurring idiomatic constructions were set to *false*.

In this way, we could have missed some idiomatic expressions which were in fact present in sentences but did not occur in a subsequent list of words. This could happen if the writer paraphrased the idiomatic expression or used it in a different case of person. However, simply allowing the flexion and the gaps in the multiword expression could heavily over-generate and introduce positive errors.

Together, we obtained 118 idiomatic features describing 118 idiomatic expression from VALLEX.

6.5 Animacy features

Animacy is a grammatical category of nouns and pronouns specifying if/how a noun or a pronoun is alive. The Wikipedia.org defines animacy as follows:

> Animacy is a grammatical category, usually of nouns, which influences the form a verb takes when it is associated with that noun.
>
> Usually, animacy has to do with how alive or how sentient a noun is. In general, personal pronouns have the highest animacy, the first-person being the highest among them. Other humans follow them, and animals, plants, natural forces such as winds, concrete things, and abstract things follow in this order; however, according to the spiritual beliefs of the people whose language possesses an animacy hierarchy, deities, spirits, or certain types of animal or plant may be ranked very highly in the hierarchy.

On contrary with this definition, we do not consider different levels of animacy but rather look at the animacy only as a boolean category. We consider a noun or a pronoun to be **animate** if and only if it is an animal (including human being).

The introduction of the animacy features was based on an assumption that animacy can often suggest the meaning of the verb. This assumption follows from the fact that some senses of verbs can only describe a relation between (human) beings.

The main problem related to the animacy features is the difficulty of the determination of animacy. There is no simple way how to determine animacy automatically, and we can only predict it for specific cases. The algorithm we used for partial animacy resolution differs for nouns and pronouns.

6.5.1 Nouns

Firstly, the Czech morphological tagger ([Hajič, 2000]) gives additional semantic information for some lemmas (especially names). This information is not part of the lemma identifier itself, it follows an underscore sign (_) in the lemma. The additional information can bear one of seven possible values:

G	geographical name
Y	given name
S	surname
E	name of a nationality
R	name of a product
K	name of a company
m	default name

In cases where the lemmatizer marked a lemma as a given name, a surname, or as a name of a nationality, we set the animacy to *true*.

Secondly, we used the morphological category *gender* which distinguishes between masculine animate and masculine inanimate in some cases, as the masculine behave differently for animate and inanimate nouns.

Thirdly, we used a list of 5,451 personal or nationality names. If a lemma is found on the list, it is also considered animate.
However, for common feminine and neuter nouns we still could not determine the animacy satisfactorily.

6.5.2 Pronoun

The morphological category *detailed part of speech* [Hajič et al., 2006] gives information about the type of the pronoun. Some types of pronouns imply animacy, others imply inanimacy.

The following list gives the values of the category *detailed part of speech*[1] which are considered as animate pronouns:

[1] Definitions are taken over from the morphological manual of the Prague Dependence Treebank [Hajič et al., 2006]

Detailed POS	Explanation
5	The pronoun *he* in forms requested after any preposition (with prefix *n-*: *něj*, *něho*, ..., lit. *him* in various cases)
6	Reflexive pronoun *se* in long forms (*sebe*, *sobě*, *sebou*, lit. *myself / yourself / herself / himself* in various cases; *se* is personless)
8	Possessive reflexive pronoun svůj (lit. *my/your/her/his* when the possessor is the subject of the sentence)
9	Relative pronoun *jenž*, *již*, ... after a preposition (*n-*: *něhož*, *niž*, ..., lit. *who*)
H	Personal pronoun, clitical (short) form (*mě*, *mi*, *ti*, *mu*, ...); these forms are used in the second position in a clause (lit. *me*, *you*, *her*, *him*), even though some of them (*mě*) might be regularly used anywhere as well
K	Relative/interrogative pronoun *kdo* (lit. *who*), incl. forms with affixes *-ž* and *-s*
P	Personal pronoun *já*, *ty*, *on* (lit. *I*, *you*, *he*) (incl. forms with the enclitic *-s*, e.g. *tys*, lit. *you're*); gender position is used for third person to distinguish *on/ona/ono* (lit. *he/she/it*), and number for all three persons

The following list gives the values of the category *detailed part of speech* which are considered as inanimate pronouns:

Detailed POS	Explanation
E	Relative pronoun *což* (corresponding to English which in subordinate clauses referring to a part of the preceding text)
Q	Pronoun relative/interrogative *co*, *copak*, *cožpak* (lit. *what*, *isn't-it-true-that*)
Y	Pronoun relative/interrogative *co* as an enclitic (after a preposition) (*oč*, *nač*, *zač*, lit. *about what*, *on/onto what*, *after/for what*)

Moreover, if the morphological category *person* of a pronoun is set to 1 or 2, the pronoun is considered as animate too.

Again, not all cases can be determined in this way.

6.5.3 Feature description

We introduced seven boolean features, one for each case, stating whether there is an animate noun or pronoun[2] in this case syntactically dependent on the verb. Moreover, we introduced another seven boolean features, one for each case, stating whether there is an animate noun or pronoun in this case anywhere in the sentence.

In cases where we could not decide about anymacy, we set the feature to *false*.

The later features do not give much detailed information about the verb. On the other hand, they can operate even in case of a wrong result of the syntactic parser.

Together we obtained 14 features for animacy.

6.6 WordNet features

In some cases, dependency of a certain lemma or a certain type of lemma on the verb can imply a particular sense of the verb. From this perspective, it might be useful to capture the presence of each lemma among the nodes dependent on the verb. However, storing the presence for all possible lemmas would lead to a huge number of features, to a loss of generality, and possible over-fitting.

There are several possibilities of how to deal with this issue. One of them is, instead of capturing presence of each and every lemma, capturing only the "class" of the lemma. This class should generalize the meaning of each word, so words with a similar meaning should belong to the same class. This solution requires usage of some kind of ontology which maps the lemmas or meanings (disambiguated lemmas) to the classes.

WordNet [Fellbaum, 1998] seemed to be a good choise for this purpose. To define a system of coarse-grained classes of WordNet items (synsets[3]), we used the WordNet top ontology designed at the University of Amsterdam [Vossen et al., 1998]. This ontology is described as a tree-based system of WordNet synsets which represents the top of the WordNet hierarchy. The ontology containing 64 items (synsets) is shown in Figure 6.7.

Using hyperonymy relation defined in WordNet we can easily determine all classes to which a given noun belongs, i.e. is related by the transitive

[2] Noun or pronoun for which we determined animacy using our limited procedure described above.

[3] The term **synset** is used in the WordNet for a lexicon item capturing single meaning. One lemma can belong to more synsets (suggesting different meaning of the lemma), as well as one synset can consist of more lemmas. See 5.1.2 for a detailed explenation

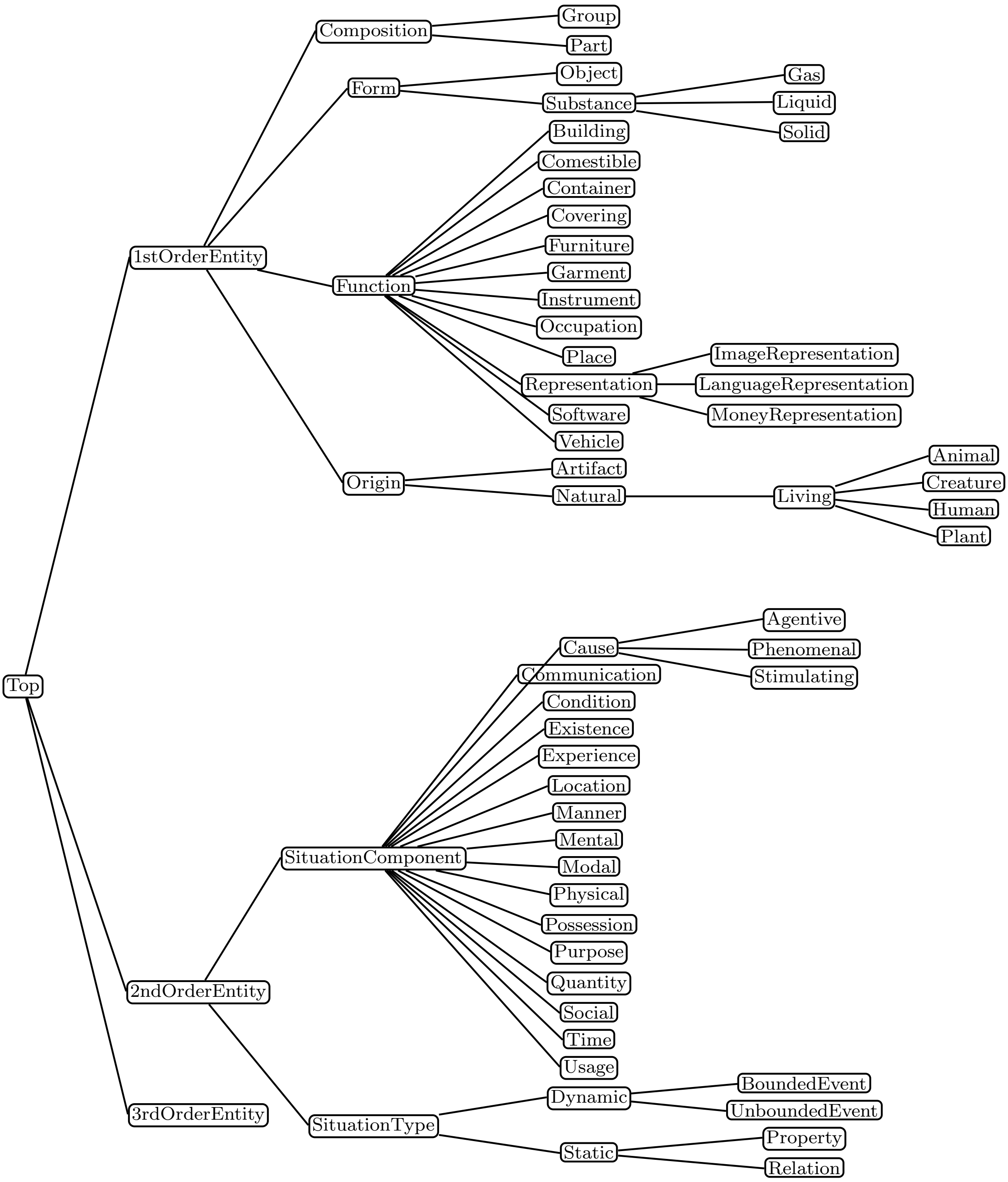


Figure 6.7: WordNet top-level ontology

relation of hyperonymy. This means that "the noun is type/kind of the class". Because of the transitivity of the hyperonymy relation, if a word belongs to a given class, it also belongs to all classes which are governing this class in the top-ontology.

6.6.1 Combination with Czech WordNet

For each lemma present in the synsets of the top ontology, we used the WordNet **Inter-Lingual-Index** to map the English WordNet to the Czech EuroWordNet [Pala and Smrž, 2004], extracting all Czech lemmas belonging to the top level classes. After this step we ended up with 1564 Czech lemmas associated to the WordNet top-level classes. As we worked with lemmas, and not with synsets, one lemma could have been mapped to more top-level classes. Moreover, if a lemma is mapped to a class, it also belongs to all its predecessors.

In the second step, we used the relation of **hyperonymy** in the Czech WordNet to determine the top-level class for the other nouns as well. We followed the relation of hyperonymy transitively until we reached a lemma assigned in the first step. Again, as we worked with the lemmas instead of synsets, one lemma could have been mapped to more top-level classes.

For each top-level class we created one feature stating whether a noun belonging to this class is directly dependent on the verb, and one feature stating whether such a noun is present anywhere in the sentence.

Together we obtained 128 WordNet class features.

Chapter 7

Evaluation

This chapter summarizes the empirical results of the experiments described in this work. We ran several machine learning algorithms on two corpora using various types of features. The setting of machine learning methods is described in Section 7.1. Because of size, we used cross-validation for the VALEVAL corpus. Moreover, two different ways of counting the overall results for the VALEVAL corpus are considered. For the Prague Dependency Treebank, we presented results for two different evaluation data sets – the development test set, and the evaluation test set. We used the development test set throughout the development period and only performed the evaluation on the evaluation data set once, for the purpose of this thesis. After that, we did not modify the methods anymore.

We will use the term **base lemma** for lemma disregarding the reflexive particle *se/si*, as it was introduced in Section 4.2.2. The lemma of a reflexive verb (e.g. *dát si*) consists of two parts, the base-lemma (*dát*) and the reflexive particle (*si*). However, as we take plain text as the input, we can not automatically distinguish between the verbs with the same base-lemma and different reflexive particle (*dát*, *dát si*, and *dát se*), so we technically consider all senses of those verbs as different senses of their base lemma. Where we use the term *lemma* in this section, it will refer to *base lemma* unless stated otherwise.

7.1 Machine Learning

Once feature vectors are generated, we can train the machine learning methods on the labeled data. The methods can later be used to perform the classification task on unlabeled data.

Chapter 2 describes different machine learning methods in general. In this section, we describe the application of concrete implementations of the methods in our task and we outline the process of training them.

The definition of senses is different for each individual verb and so are the classification tasks for the verbs, therefore the classifier has to be trained for each verb separately. For corpora with a real text distribution (the Prague Dependency Treebank), the number of samples in the running text is small for most of the verbs, following the Zip's law, so the methods are often trained on few training samples only which might influence their performance in a negative manner.

For the VALEVAL corpus, where the verb distribution does not reflect the real text distribution, this is not an issue. However, for this corpus, the absolute number of data samples is low, which made us use the 10-fold cross-validation for more reliable evaluation.

We tested several classification methods to find which classifier best suits the disambiguation task. The methods included the Naïve Bayes classifier, two different implementations of decision trees, rule-based learning, the maximum entropy model, and support vector machines. Some tools implementing the classifiers have special demands on the features format, so the features had to be modified, otherwise only some of the features could have been used.

7.1.1 Naïve Bayes Classifier

The Naïve Bayes Classifier is a simple probabilistic classifier using features independence assumption. We did not expect this classifier to outperform other more state-of-art classifiers, however, our goal was to measure how much better the other classifiers work. This classifier is easy to implement. If the other more sophisticated classifiers do not outperform it significantly, the Nav̈e Bayes Classifier will be an interesting choice due to its simplicity.

We used Christian Borgelt's implementation[1] of the Naïve Bayes Classifier. It is an open-source implementation written in C using command-line interface and supporting comma-separated-values data files. The implementation allows us to train the classifier on a data portion (training file) and save the classifier model to a file. Later, the saved classifier can be used to classify data samples from a testing portion (testing file). The testing file has a similar data format to the training file, it is only missing the class attribute. The tool does not provide any evaluation outputs but the predicted class is assigned.

[1] http://fuzzy.cs.uni-magdeburg.de/∼borgelt/bayes.html

7.1.2 Decision Trees

For testing the decision trees we used two different implementations. The first is a commercial tool and the other is an open-source.

C5 Implementation

The C5.0 toolkit is a commercial toolkit developed by an Australian company, RuleQuest Research[2]. It implements an algorithm C5.0, which is currently not published.

The C5.0 algorithm is basically a modification of the C4.5 algorithm, it is its ancestor, only with minor changes. The C4.5 also used to be a commercial package, however it became free when the authors published the new version of the toolkit. Nowadays, not only is the C4.5 toolkit free, but the algorithm is also published and there are other implementations too. Compared to C4.5, C5.0 provides additional features, of which the most important for our purpose is support of cross-validation, automatic winnowing of attributes, and implementation of rule-based learning.

Borgelt's Implementation

The second decision trees algorithm which we used was Christian Borgelt's implementation[3].

Like the Naïve Beyes implementation, it is also an open-source implementation written in C, using the same input data formats as the Naïve Beyes implementation and same mechanism for the data processing holds.

7.1.3 Rule-based Learning

The rule-based learning which we used is the one implemented in the C5.0 toolkit (7.1.2). The results of the decision trees and the rule-based methods are strongly correlated as C5.0 derives the rules from the decision trees. Still, it is not a straightforward transcription into rules; the rule-based methods are different classifiers and could perform differently, according to the author's statement.

7.1.4 Maximum Entropy Model

For the Maximum Entropy Model, we used the Mallet toolkit implementation [McCallum, 2002]. Mallet is a toolkit implementing several methods,

[2] http://www.rulequest.com/

[3] http://fuzzy.cs.uni-magdeburg.de/∼borgelt/dtree.html

including Naïve Beyes, Maximum Entropy, Conditional Random Fields, and others. It is implemented in Java language. The toolkit can be used for the classification of documents (provided as plain text documents), or lists of features in a comma-separated-values file. The Mallet toolkit supports cross-validation but we used our own mechanism instead.

7.1.5 Support Vector Machine

As the Support Vector Machine (SVM) classification we used the **e1071**[4] package of the statistical environment **R**. This package is one of the most important R packages, and contains several machine learning and statistical methods. The implementation of SVM supports different kernels. In our experiments we used the linear, polynomial, radial, and sigmoid kernels. Speed is one of the big issues of this implementation, probably because of the R-platform. Neither the package itself nor the SVM supports cross-validation, therefore we used our own work-around utility in the R language.

7.2 Result Weighting

The VALEVAL corpus contains verbs selected from the VALLEX lexicon by humans, and every selected word includes the same number of samples, regardless of their relative frequency in the language. However, if we want to claim something about the performance of a particular method on real data, we intuitively expect that more common verbs should be cared about more than rearer ones. Therefore, combining results for individual verbs with flat weights (corresponding to their flat frequencies in the VALEVAL corpus) does not give what we might expect.

For this reason, we performed two different types of evaluation. In the first one, we computed the average of the results for individual lemmas weighted by the frequencies in the corpus, but in the second one, we weighted the results by the relative frequencies measured in the Czech National Corpus. The Czech Nation Corpus (CNC) [Kocek et al., 2000] is a large Czech corpus containing more than 100 million of words and a diverse scale of genres.

Weighting results by the relative frequencies in the CNC gives information about how well the method would perform on real data (supposing that the distribution of the CNC is close enough to the data which we call *real*). But it does not tell much about the quality of the method, as it can suffer from the wrong selection of verbs. On the other hand, weighting by the number

[4] http://cran.r-project.org/src/contrib/Descriptions/e1071.html

of sentences in the data makes more sense when comparing the performance of the methods but it does not claim anything about the real performance.

7.3 Baseline

As the baseline of the disambiguation task we took **the relative frequency of the most frequent frame of each lemma in the training data**.

For the VALEVAL corpus, we determined the baseline using 10-fold cross validation. The baseline for each fold was weighted by the number of samples in it, and the arithmetic average was computed.

For the Prague Dependency Treebank, the baseline was measured on the testing data (the dtest, and the etest section respectively) but the most frequent frame was stated from the training data.

The baselines for individual verbs started on 23.81% for the VALEVAL corpus (lemma *vzít* with 10 different annotated frames). For the PDT corpus, the baseline was zero for 172 verbs as no frames from the testing data set occurred in the training data. The maximal baseline was 100% for verbs with only one frame.

A high baseline indicates that there is a dominant sense of the corresponding verb, to which a high portion of running verbs belong. A low baseline, on the other hand, indicates that the senses of the verb are more spread and there are more senses of the verb which are common in the corpus.

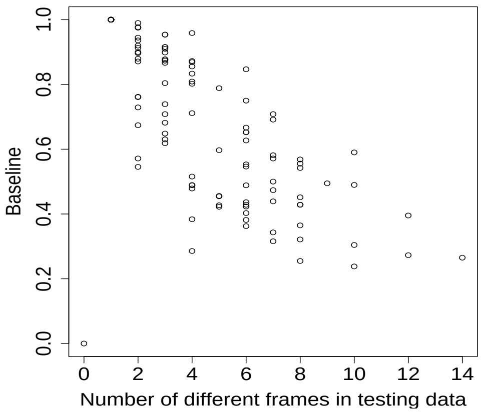


Figure 7.1: Baseline depending on the number of frames for the VALEVAL corpus.

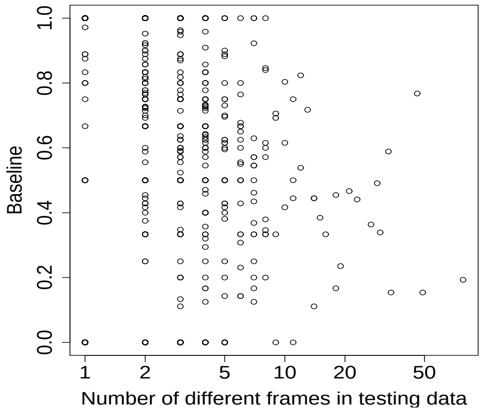


Figure 7.2: Baseline depending on the number of frames for the PDT.

In the VALEVAL corpus, the baseline for particular verbs is highly correlated with the number of different frames in the training data set, as Figure 7.1 shows. However, the correlation is much lower in the case of the PDT (see Figure 7.2). This is because the PDT contains many verbs with few samples which breaks the dependency. In the VALEVAL corpus, all verbs have a comparable number of samples.

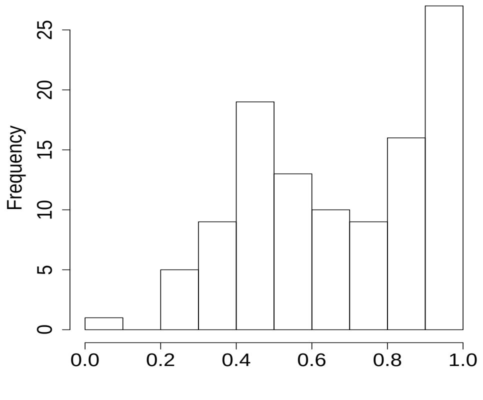


Figure 7.3: Distribution of the relative frequency of the most frequent frames for the VALEVAL corpus.

Figures 7.3 and 7.4 show the histograms of the relative frequencies of the

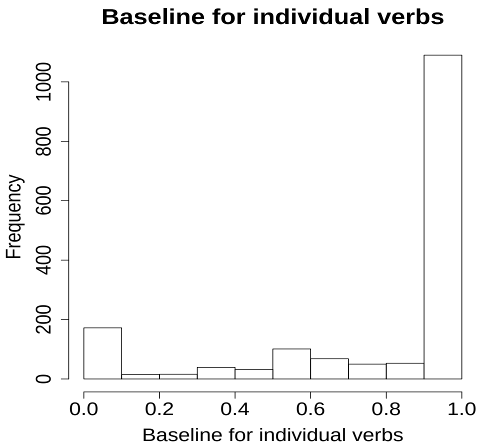


Figure 7.4: Distribution of the relative frequency of the most frequent frames for the PDT.

most frequent frames (the baselines). The horizontal axis gives the baseline value split into intervals, and the corresponding vertical value gives the number of verbs with a baseline from this interval.

In the VALEVAL corpus, there is a considerable amount of words with a baseline close or equal to 100% (a single frame), another “concentration of baselines” is around 50%. In the PDT, the number of lemmas with baseline close or equal (equal, in fact) to 100% is even higher, which follows from the high number of lemmas with a small number of samples. Another concentration of baselines is close to zero (zero, in fact), which also includes verbs with a small number of samples (usually two or three) but with different frames. The data for PDT was generated from the development testing set.

We computed the overall baseline as the weighted average of the individual baselines. The overall baseline for the VALEVAL corpus was 68.27% when weighted by the number of sentences in our data set and 60.74% when weighted by the relative frequency in the Czech National Corpus. The overall baseline for PDT was 73.19% for the development testing set and 71.98% for the evaluation testing set. The baseline statistics are summarized in Table 7.1.

7.4 Results

This section presents the evaluation results of the valency frame disambiguation using each presented type of features separately, as well as different combinations of feature types, computed by different classifiers.

	VALEVAL		PDT	
	$\oslash_{data}$	$\oslash_{CNC}$	**dtest**	**etest**
Average number of frames	4.45	5.31	2.39	2.27
Baseline	68.27	60.74	73.19	71.98

$\oslash_{data}$ denotes average weighted by the number of sentences in the dataset.
$\oslash_{CNC}$ denotes average weighted by the number of sentences in the Czech National Corpus.

Table 7.1: Difficulty of the frame disambiguation task

Table 7.2 presents the results on the VALEVAL corpus obtained by weighting the individual results with the number of samples in the corpus, while Table 7.3 shows the results weighted by the relative frequencies in the CNC.

Table 7.4 and Table 7.5 present the results for the Prague Dependency Treebank for development and evaluation testing set respectively.

The columns of the tables correspond to different classification methods: Naïve Bayes classifier (NBC), Christian Borgelt's implementation of the decision trees (DTREE), C5 decision trees (C5-DT), and C5 rule-based learning (C5-RB), Support Vector Machines (SVM), and Maximum Entropy (ME). The rows of the table correspond to different types of features, the first five rows state the results when using each type of features separately, the following rows state the results for different combinations of the type.

The best accuracy on VALEVAL (CNC weighting) – 77.56% – was achieved by the C5 rule-based algorithm using the full set of features. The best accuracy on PDT (evaluation testing set) – 78.88% – was achieved by the Support Vector Machines using the syntax-based and idiomatic features.

We calculated **oraculum** for all the tasks as the accuracy when the correct class is always chosen if it has been seen in the training data. Only data samples from classes which has not occured in the training data are counted as errors. The oraculum states the upper limit of the accuracy for the given data.

The oraculum was high for all the data sets. For VALEVAL, it was 98.78% and 95.32% when weighted by the data frequencies and the frequencies in CNC, respectively. For PDT, the oraculum was 98.08% and 97.75% on the development and evaluation testing data, respectively.

Corpus:	**VALEVAL**					
Weighting:	**Sample counts in the corpus.**					
Type of features	**NBC**	**DTREE**	**C5-DT**	**C5-RB**	**SVM**	**ME**
Baseline	68.27					
Morphological (M)	71.77	70.33	73.65	73.94	69.54	73.77
Syntactical (S)	76.96	77.21	78.22	78.24	78.05	77.55
Animacy (A)	65.78	68.18	70.76	70.90	69.97	68.52
Idiomatic (I)	68.17	68.26	68.37	68.39	68.45	67.23
WordNet (W)	62.90	66.50	70.52	70.51	66.11	65.74
M + S	73.39	71.28	78.71	78.56	73.87	77.26
M + I	71.68	70.33	73.81	74.00	68.58	75.19
S + W	73.67	74.89	78.72	78.79	74.94	75.57
S + A	73.42	71.24	78.25	78.70	76.09	76.84
S + I	77.03	77.48	78.37	78.35	78.19	78.00
M + S + I	73.36	71.28	78.97	78.74	73.67	77.16
M + S + A	74.41	71.00	79.20	79.19	73.73	78.04
M + S + W	74.20	71.06	79.42	79.22	74.04	77.18
S + A + W	72.86	71.15	78.97	79.37	74.58	77.02
S + A + I	73.43	71.24	78.40	78.65	76.01	77.21
S + I + W	73.98	75.04	78.63	78.98	75.07	75.82
M + S + I + W	74.08	71.06	79.20	79.39	74.13	77.48
M + S + A + W	74.51	70.84	79.54	79.74	74.65	78.32
S + A + I + W	72.97	71.15	79.31	79.29	74.54	77.17
M + S + A + I + W	74.47	70.84	79.66	**80.10**	74.89	77.74

Results are obtained by weighting individual results with the relative frequencies in the VALEVAL corpus.

Table 7.2: Accuracy [%] of the frame disambiguation task for VALEVAL corpus.

7.5 Methods Comparison

This section compares the classification methods and discusses their appropriateness with regard to the disambiguation task.

Different methods achieved different results on different data. Generally, we can claim that the C5 decision trees, C5 rulesets, Support Vector Machines and the Maximum Entropy model achieved comparably good results throughout the experiments. As has already been mentioned, we did not expect the Naïve Bayes classifier to beat other state-of-art methods. The second implementation of the decision trees algorithm (DTREE) also did not achieve results comparable with C5.

Corpus:	**VALEVAL**					
Weighting:	**Relative frequencies in the Czech National Corpus**					
Type of features	**NBC**	**DTREE**	**C5-DT**	**C5-RB**	**SVM**	**ME**
Baseline	60.74					
Morphological (M)	61.62	59.81	67.50	67.83	58.48	66.36
Syntactical (S)	69.98	69.34	71.01	70.43	67.90	68.51
Animacy (A)	52.87	59.86	62.32	62.67	55.12	59.60
Idiomatic (I)	60.89	60.21	61.01	61.10	60.96	62.77
WordNet (W)	45.32	53.62	58.34	59.22	50.72	54.30
M + S	63.52	60.25	69.69	69.15	63.34	64.11
M + I	61.65	59.81	67.77	68.40	58.61	63.65
S + W	59.37	60.85	71.28	70.87	60.60	61.70
S + A	63.44	61.67	70.56	70.56	63.96	63.26
S + I	69.42	69.61	70.96	70.55	68.03	69.95
M + S + I	63.52	60.25	69.27	68.54	63.43	68.76
M + S + A	63.13	58.19	69.91	69.46	64.39	64.74
M + S + W	64.80	60.28	76.61	75.08	65.27	62.62
S + A + W	60.68	61.43	70.65	71.07	58.75	65.05
S + A + I	63.32	61.67	70.95	71.31	64.04	67.22
S + I + W	59.63	60.94	71.10	71.23	61.57	65.84
M + S + I + W	64.78	60.28	76.90	77.25	65.30	63.62
M + S + A + W	64.59	58.36	76.85	77.10	62.62	67.51
S + A + I + W	60.78	61.43	71.33	71.31	58.67	64.65
M + S + A + I + W	64.58	58.36	76.97	**77.56**	62.64	67.45

Results are obtained by weighting individual results with the relative frequencies in the Czech National Corpus.

Table 7.3: Accuracy [%] of the frame disambiguation task for VALEVAL corpus.

The **C5** algorithm proved to be a reliable classification method. Compared to other methods, it performed well even if the number of training samples was low. When the number of samples was higher, the Maximum Entropy models tended to outperform C5.

C5 decision trees and rule-sets are comparably powerful, sometimes one scores slightly better, sometimes the other one does. The differences are usually not significant. Still, the rule-sets seemed to work slightly better in our tasks, which corresponds to the statement of the C5's authors. On the PDT evaluation test set, both C5 algorithms achieved the same result (78.06%).

The C5 method showed some gain even with very poor feature sets (ani-

Corpus:	**PDT - dtest**					
Weighting:	**Sample counts in the corpus.**					
Type of features	**NBC**	**DTREE**	**C5-DT**	**C5-RB**	**SVM**	**ME**
Baseline	73.19					
Morphological (M)	74.42	75.26	75.86	75.82	74.27	76.58
Syntactical (S)	78.59	78.75	77.88	77.79	79.23	79.35
Animacy (A)	71.61	72.82	73.59	73.59	73.64	73.30
Idiomatic (I)	73.77	73.71	73.49	73.48	73.78	73.60
WordNet (W)	68.97	71.53	73.25	73.28	71.59	71.82
M + S	76.31	76.13	78.82	78.91	78.52	79.06
M + I	74.39	75.31	75.97	76.10	74.73	76.68
S + W	76.05	77.41	77.91	77.95	77.72	78.20
S + A	76.66	75.73	77.96	77.83	78.25	78.43
S + I	79.15	79.23	78.29	78.21	**79.76**	79.46
M + S + I	76.28	76.23	79.15	79.26	78.82	79.15
M + S + A	76.28	75.94	78.37	78.37	77.59	78.97
M + S + W	76.58	76.01	78.40	78.57	77.98	79.39
S + A + W	76.09	74.99	78.13	78.10	76.74	78.05
S + A + I	77.43	75.97	78.52	78.38	78.52	78.97
S + I + W	76.04	77.82	78.29	78.44	78.05	78.45
M + S + I + W	76.43	76.10	78.81	78.88	78.29	79.42
M + S + A + W	76.29	75.93	78.30	78.42	78.05	79.42
S + A + I + W	76.25	75.15	78.46	78.60	77.13	78.60
M + S + A + I + W	76.47	76.02	78.66	78.82	78.23	79.67

Table 7.4: Accuracy [%] of the frame disambiguation task for the development test set of the Prague Dependency Treebank.

macy or idiomatic features alone), compared to other methods which usually scored below the baseline. As a matter of fact, the C5 methods (with features winnowing) never scored worse than the baseline, which does not hold for any other method examined.

The winnowing of the feature space before the actual execution of the classifier usually does not hurt, on the contrary, it often helps. Tables 7.6 and 7.7 show the results of the C5 decision trees with and without winnowing of the features for different combinations of features.

The winnowing improves the results more for the VALEVAL corpus where the average number of samples is lower. With more features, the need for winnowing increases, which agrees with the intuition.

Corpus:	**PDT - etest**					
Weighting:	**Sample counts in the corpus.**					
Type of features	**NBC**	**DTREE**	**C5-DT**	**C5-RB**	**SVM**	**ME**
Baseline	71.98					
Morphological (M)	73.03	73.72	73.66	73.62	72.55	74.59
Syntactical (S)	77.84	77.89	77.47	77.35	78.63	78.60
Animacy (A)	70.23	71.05	72.37	72.37	71.99	71.44
Idiomatic (I)	72.45	72.26	72.49	72.49	72.59	72.35
WordNet (W)	68.04	70.41	72.14	72.09	70.15	70.58
M + S	75.24	75.18	77.48	77.54	76.78	78.06
M + I	73.30	73.73	73.66	73.73	72.82	74.89
S + W	74.89	76.43	77.66	77.50	76.35	76.85
S + A	76.19	74.22	77.51	77.40	77.19	77.70
S + I	78.17	78.15	77.76	77.66	**78.88**	78.85
M + S + I	75.18	75.22	77.71	77.80	76.89	78.10
M + S + A	75.52	75.09	77.25	77.33	75.75	78.09
M + S + W	75.72	74.97	77.60	77.75	76.46	78.17
S + A + W	75.12	73.61	77.00	76.93	75.37	76.89
S + A + I	76.45	74.38	77.75	77.61	77.42	78.04
S + I + W	74.98	76.68	77.80	77.66	76.56	76.95
M + S + I + W	75.79	75.00	78.06	78.06	76.70	64.48
M + S + A + W	75.67	75.10	77.74	77.76	75.93	78.00
S + A + I + W	75.35	73.74	77.57	77.50	75.51	77.07
M + S + A + I + W	75.51	75.13	77.91	78.04	76.10	78.26

Table 7.5: Accuracy [%] of the frame disambiguation task for the evaluation test set of the Prague Dependency Treebank.

Corpus:	**VALEVAL**	
Type of features	**without winnowing**	**with winnowing**
M	72.80	73.65
S	77.71	78.22
A	69.37	70.76
I	68.46	68.37
W	67.98	70.52
M + S	77.08	78.71
M + W	70.98	73.76
M + S + A + W	76.78	79.54
M + S + A + I + W	76.68	79.66

Table 7.6: Winnowing features for C5 decision trees – VALEVAL

Corpus:	**PDT - etest**	
Type of features	**without winnowing**	**with winnowing**
M	73.67	73.66
S	78.14	77.47
A	71.98	72.37
I	72.54	72.49
W	71.16	72.14
M + S	77.67	77.48
M + W	73.31	73.62
M + S + A + W	77.71	77.74
M + S + A + I + W	77.71	77.91

Table 7.7: Winnowing features for C5 decision trees – PDT evaluation testing set

Support vector machines is a popular classifier which is in general performing well. However, it requires a fine tuning of the parameters.

In our experiments, the linear kernel always scored best. This can be explained by the fact that we largely used boolean features which could be easily separated by a superspace in the linear space. Using a more sophisticated kernel adds freedom in the methods which makes the classifier more difficult to train. If there were more real-number features, the situation would probably differ. However, linguistic characteristics are rarely described by real-number features.

The support vector machines achieved the absolutely best result on both, the development and the evaluation testing dataset of the Prague Dependency Treebank.

Due to the categorical nature of morphological features, the SVM used their modified (booleanized) version, where each value of each morphological feature created a new feature stating whether the feature has the corresponding value. Instead of the original 60 features, we used 705 booleanized morphological features.

7.6 Features Comparison

This section gives comparison of individual types of features.

Tables 7.2 through 7.5 show that the syntax-based features (see Section 6.3) clearly performed best in all datasets. They contain most of the information which is linguistically relevant to the valency.

The morphological features turned out to be the second best. The strong difference between syntax-based and morphological features shows how much the statistical parsing helps to analyze the meaning of the verbs. The remaining feature types achieved similar results, usually in the following order: idiomatic features, animacy features, WordNet features.

When we look at the combination of syntax-based features with another type of features, the best result was achieved with the idiomatic features, while the combination with morphological features usually performed worst. In our opinion, this is because the information stored in the morphological features is already included in the syntactic features and adding it does not bring any new information. On the other hand, the other types of features contain information of a different kind, hence they help the syntactic features when combined.

7.6.1 Differences in Words

The success of the disambiguation task is not flat across all the verbs, it differs from one verb to another as differ the verb's characteristics. The most of the verbs have a single dominant sense which is annotated to the majority of the running verbs. Typical examples are the verbs *být* (the most frequent Czech verb), *říci* or *začít*. There are, however, other verbs, whose different senses are widely spread and used in the language. Typical examples are the verbs *mít* (the second most frequent Czech verb), *dát*, or *vědět*.

In the following sections, we present decision trees generated by the C5 algorithms. We chosen decision trees because it is a white-box model, so they clearly show how the classifier works. Rule-based methods are in fact another form of serializing the decision trees, and the differences are small.

VALEVAL

Figure 7.5 shows 50 verbs selected from VALEVAL sorted by the relative frequency of their most frequent frame in the corpus (the baseline). For each verb, the graph shows the portions of data annotated to different frames in different colors (hues).

The C5 decision trees scored worse than the baseline for eight verbs in the VALEVAL corpus. The following table lists the verbs with possible explanations of the fails:

zachytnout	(29 % loss)	low number (7) of training samples (4 frames)
spojit	(3 % loss)	high number (6) of frames
držet	(3 % loss)	high number (8) of frames
přidat	(2 % loss)	high number (7) of frames
ponechávat	(1 % loss)	
stávat	(1 % loss)	

Figure 7.6 shows the decision tree for the verb *stávat*, the decision trees for the other verbs from the previous list are not interesting.

The verbs with the highest performance gain ($accuracy - baseline$) were the following:

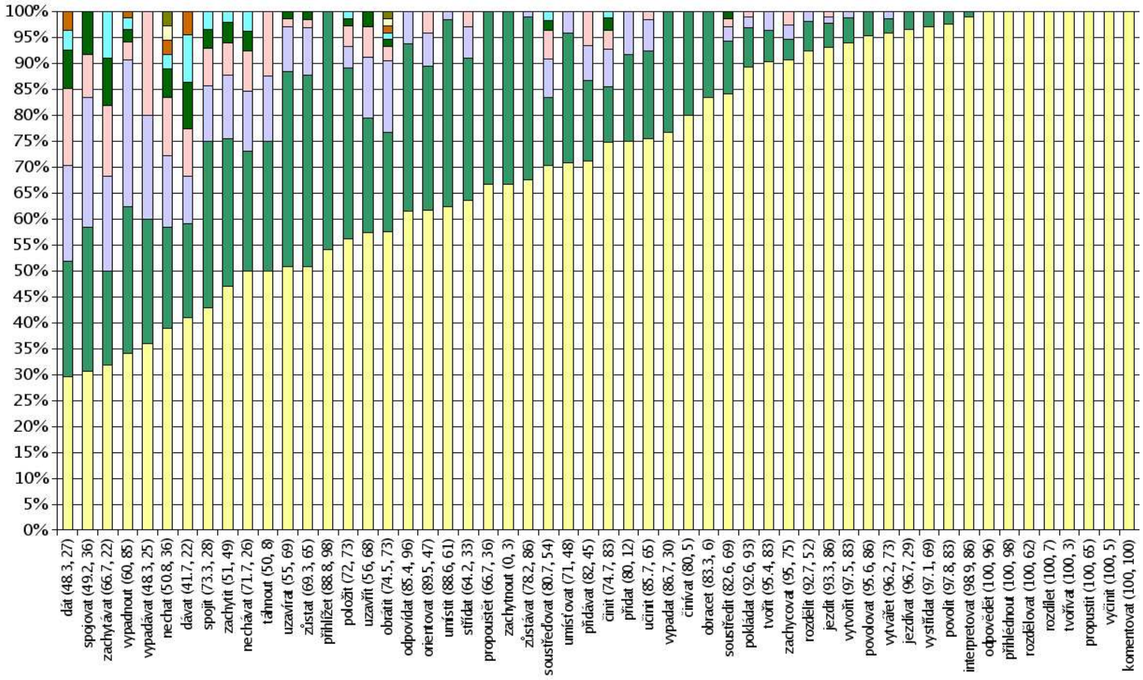
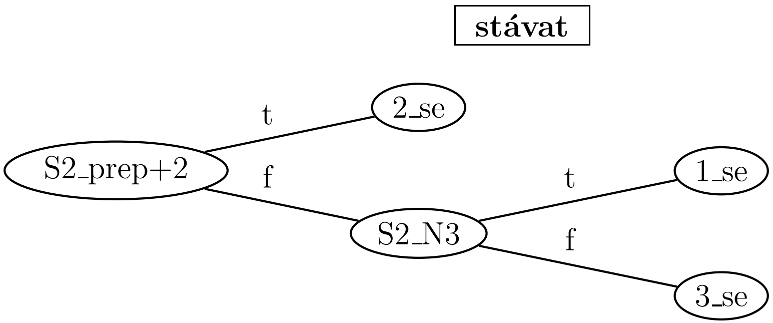


Figure 7.5: Number of annotated running verbs for individual verbs (sorted by the number of most frequent frame).

odebrat	(48 % gain)
stát	(43 % gain)
určit	(35 % gain)
přihlížet	(33 % gain)
vyvíjet	(32 % gain)
udržovat	(31 % gain)
připadnout	(31 % gain)
orientovat	(31 % gain)
dát	(31 % gain)
umístit	(30 % gain)
vyvinout	(30 % gain)
přiznat	(30 % gain)

Figures 7.7 and 7.8 show the decision trees for the verb *odebrat* and *udržovat* respectively.



S2_prep+2 . . . presence of a preposition in genitive dependent on the verb
S2_N3 . . . presence of a dative noun dependent on the verb

1_se	přiházet se; uskutečňovat se • často se mi stávalo, že jsem přišel pozdě
2_se	přeměňovat se • pomalu se z něj stávala příšera
3_se	přeměňovat se v něco • z chlapce se stával mužem

Figure 7.6: Decision tree for the verb *stávat* from VALEVAL.

PDT

The C5 decision trees scored worse than the baseline for 64 verbs out of 1712. The verbs with the lowest performance were the following:

znát, *držet*, *učinit*, *přijímat*, *předpokládat*, *růst*, *fungovat*, *vyhrát*, *přinést.*

The most often reason for the fails were a low number of training data (unreliable classifier) or testing data (unreliable result), high number of frames compared to the size of training data (e.g. verb *držet* – 18 frames for 55 running verbs) and inability to distinguish two frames (e.g. for the verb *získat* the classifier did not distinguished frames v-w9501f1 /*vydolovat, dostat, obdržet, vylákat*/ and v-w9501f2 /*naklonit si, vydobýt*/ correctly).

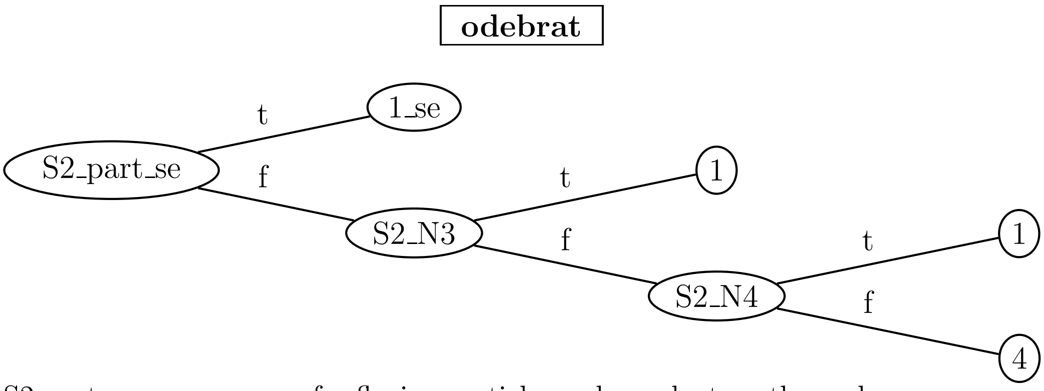
The verbs with the highest positive influence on the total performance (*accuracy* – *baseline*) were the following (in this order):

být, *mít*, *stát*, *dostat*, *rozhodnout*, *myslit*, *dát.*

Figures 7.9 and 7.10 show examples of decision trees for the verbs *rozhodnout* and *dělit*, respectively.

Appendix B shows other selected decision trees generated from the training data-set of PDT by the C5 algorithm.

In some decision trees, overfitting due to a low number of training samples

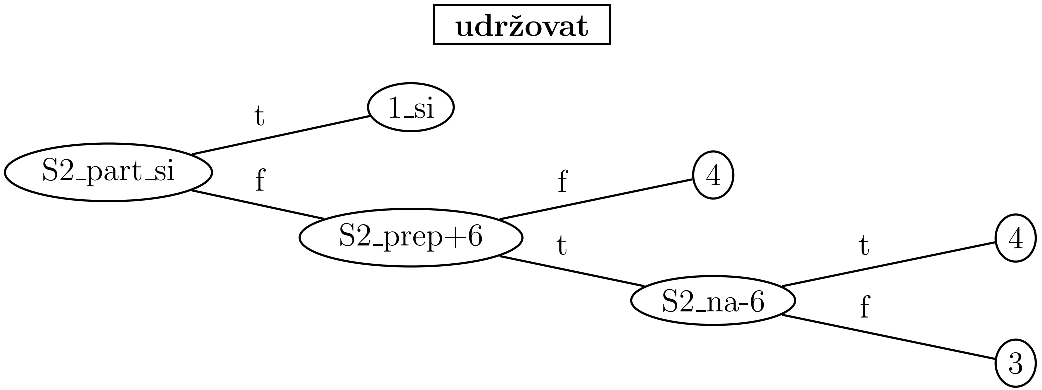


S2_part_se . . . presence of reflexive particle *se* dependent on the verb
S2_N3 . . . presence of a dative noun dependent on the verb
S2_N4 . . . presence of an accusative noun dependent on the verb

1_se	odejít; vydat se • odebral se na schůzi
1	odejmout • odebrali jí děti
4	odkoupit; převzít • odebrali všechno objednané zboží

Figure 7.7: Decision tree for the verb *odebrat* from VALEVAL.

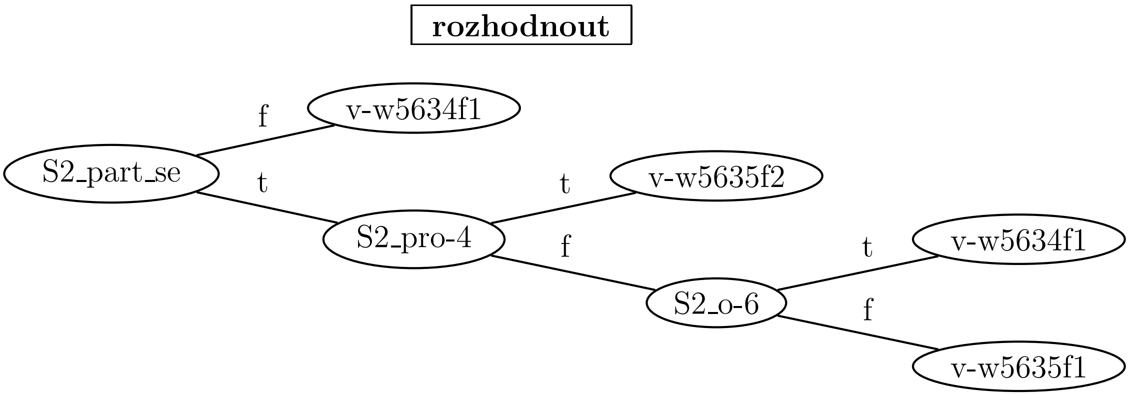
can be seen (verb *vyslovit*). In other cases, wrong morphological analysis influenced the resulting decision tree (verbs *věřit* or *žádat*). In the decision tree for verb *zavést* is apparent correct application of a WordNet feature.



S2_part_si . . . presence of reflexive particle *si* dependent on the verb
S2_prep+6 . . . presence of preposition in local dependent on the verb
S2_na-6 . . . presence of preposition *na* in local dependent on the verb

3	zachovávat v určitém stavu • udržoval byt v čistotě
4	dodržet; uchránit; pečovat • udržoval kázeň / pořádek / kontakty / zahradu
1_si	zachovávat • udržoval si nadhled / kondici

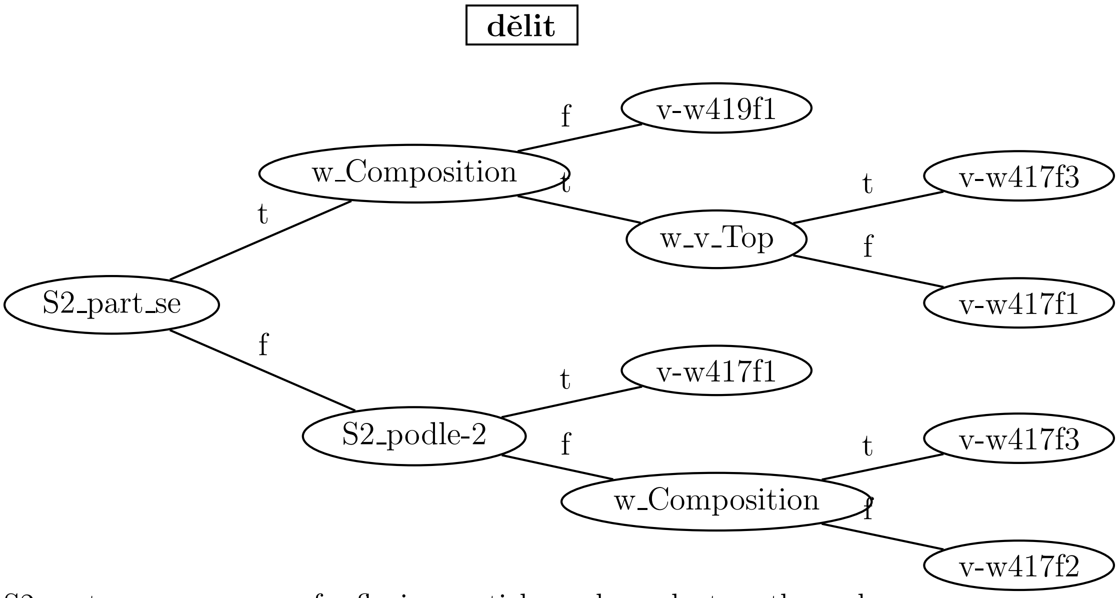
Figure 7.8: Decision tree for the verb *udržovat* from VALEVAL.



S2_part_se . . . presence of reflexive particle *se* dependent on the verb
S2_pro-4 . . . presence of preposition *pro* in accusative dependent on the verb
S2_o-6 . . . presence of preposition *o* in local dependent on the verb

v-w5634f1	určit
	• rychle rozhodl o jeho přijetí
	• r. přijmout všechny
	• r., kam půjdeme
v-w5635f1	
	• rychle se rozhodl o dalším postupu
	• r. se přijmout opatření
	• r. se, kam půjde
	• r. se rychle, jestli mu vydají....
v-w5635f2	volit, vybrat
	• rozhodnout se pro Prahu mezi dvěma možnostmi
	• r. se pro Karla

Figure 7.9: Decision tree for the verb *rozhodnout* from PDT.



S2_part_se . . . presence of reflexive particle *se* dependent on the verb
w_Composition. . . Presence of a noun from semantic class *Composition* anywhere in the sentence
w_v_Top . . . Presence of a noun from semantic class *Top* dependent on the verb
S2_podle-2 . . . Presence of preposition *podle* in genitive dependent on the verb

v-w417f1	členit, rozdělit, kouskovat • dělit příjmení na části • d. republiku na dva státy • d. salám na poloviny • d. salám nožem v polovině • d. úkol na několik etap
v-w417f2	odloučit • minuta dělila kajakářku od medaile
v-w417f3	rozdělit, dát, podělit • dělit archívy mezi republiky • dělit dětem dárky • d. mezi děti dárky • d. aktivity na střediska, do středisek, střediskům • d. peníze do rozpočtu obcí
v-w419f1	rozdělit se • dělil se s příbuznými o majetek • ODS se dělí s ČSSD o politickou moc

Figure 7.10: Decision tree for the verb *dělit* from PDT.

7.6.2 Importance of Features

To compare the impact of individual features, we observed their frequencies in the decision trees. We used the full feature set for this experiment, and we summed the occurrences of all features for all verbs in the experiment.

Following the intuition, the features used in the higher levels (the levels closes to the root) of the trees are more important for making the decisions that the features in the nodes of the lower levels, because the classification algorithm decides earlier based on the top-level features and the decision applies to more samples. To reflect this difference, instead of simple counting all features equally, we weighted the occurrence of individual features in the decision trees by the 0.5-based exponent of the level in which they occurred (1 for the root, 0.5 for the first level, 0.25 for the second level, ...). The weight w of a feature f_i was calculated as follows:

$$w(f_i) = \sum_{t \in T} log_{\frac{1}{2}} level_t(f_i) \mid f_i \text{ used in } t$$

where T is the set of all trees in the experiment and $level_t(f_i)$ gives the level in which is the feature f_i used in the tree t. Figure 7.11 provides graphical representation of the weights of different position in a decision tree.

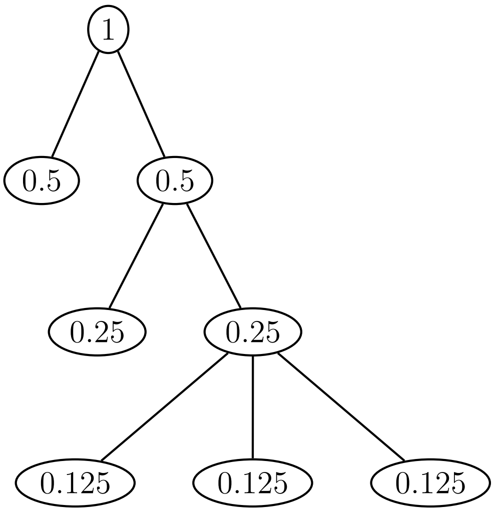


Figure 7.11: Weighting of features for computation of feature importance

In the case of the VALEVAL corpus, we summed over all the possible trees resulting from the cross-validation.

Table 7.8 and 7.9 show the features which resulted as the most important ones for the VALEVAL and PDT, respectively. The absolute values do not

Feature type	Feature description	Weight
Syntax-based	Presence of reflexive particle *se* dependent on the verb	518.0
Syntax-based	Presence of preposition in accusative dep. on the verb	253.1
Morphological	Gender of the word following the verb	253.0
Morphological	Voice of the verb	121.5
Syntax-based	Pres. of noun or a subst. pron. in dative dep. on the verb	110.4
Morphological	Gender of the verb	96.0
Morphological	Case of the word two possitions after the verb	80.0
Morphological	Part of speech of the word following the verb	75.5
Syntax-based	Presence of a verb (in infinitive) dependent on the verb	70.0
Syntax-based	Presence of preposition *za* in genitive dep. on the verb	68.0
Syntax-based	Presence of preposition in dative dependent on the verb	62.2
Syntax-based	Presence of preposition in local dependent on the verb	62.2
Morphological	Case of the word following the verb	60.9
Syntax-based	Pres. of noun or a subst. pron. in instr. dep. on the verb	58.0
Syntax-based	Presence of preposition *za* in accusative dep. on the verb	56.2
Syntax-based	Presence of reflexive particle *si* dependent on the verb	51.4
Morphological	Number of the word following the verb	47.1
Morphological	Number of the verb	41.5
Animateness	Tells wheather there is a animate substantive in genitive	41.1
Syntax-based	Presence of preposition *do* in genitive dep. on the verb	36.5
Wordnet	Presence of a noun from sem. class SituationComponent	31.4

Table 7.8: Features most often chosen in the decision trees for VALEVAL corpus

have any reasonable interpretation.

The results suggest that the syntax-based and morphological features were used most often for the important decisions in both corpora.

The absolutely most important feature was the presence of the reflexive particle *se* dependent on the verb. In Czech, the reflexive particles *se* and *si* are often used as reflexive tanta in which case they are sense marking. A verb with and without the reflexive tantum has a different meanings even if other sentence constituents remain the same. The reflexive tantum is usually considered as a part of the lemma. The reflexive particle *si* is, however, rearer compared to the particle *se*.

The second most important feature is the presence of a preposition in the accusative dependent on the verb. The accusative introduces direct object and its presence often indicates different sense of a verb.

Feature type	Feature description	Weight
Syntax-based	Presence of reflexive particle *se* dependent on the verb	162.1
Morphological	Detailed part of speech of the word preceeding the verb	136.7
Morphological	Case of the word two possitions before the verb	50.3
Morphological	Gender of the word following the verb	48.4
Morphological	Gender of the word two possitions before the verb	40.6
Syntax-based	Presence of reflexive particle *si* dependent on the verb	39.8
Morphological	Part of speech of the word two possitions before the verb	39.2
Morphological	Case of the word two possitions after the verb	39.0
Morphological	Number of the word following the verb	34.2
Syntax-based	Pres. of noun or a subst. pron. in dat. dep. on the verb	32.8
Morphological	Tense of the verb	27.9
Syntax-based	Presence of preposition *do* in genitive dep. on the verb	25.9
Syntax-based	Pres. of preposition in accusative dependent on the verb	23.8
Morphological	Gender of the word two possitions after the verb	21.8
Morphological	Degree of compar. of the word two pos. bef. the verb	20.5
Morphological	Detailed POS of the word two possitions after the verb	20.0
Morphological	Number of the word preceeding the verb	20.0
Syntax-based	Pres. of noun or a sub. pron. in nom. dep. on the verb	20.0
Syntax-based	Presence of a verb (in infinitive) dependent on the verb	18.8
Morphological	Negation of the word following the verb	18.2
Wordnet	Presence of a noun from sem. class Function	17.7
Animateness	Presence of an anim. subst. in nom. dep. on the verb	14.2

Table 7.9: Features most often chosen in the decision trees for the PDT corpus

7.6.3 Feature Overall Statistics

In the previous chapter, we measured how many times were individual features used in the C5 decision trees. Now we sum over features types to get the overall statistics. In all runs of the cross-validation on the VALEVAL corpus, 105 features were used at least once, while 318 features were not used at all.

In PDT, the number of features used in the decision trees was 200 for the development testing set, and 202 for the evaluation testing set, respectively, and the number of unused features was 223 for the development testing set, and 221 for the evaluation testing set, respectively.

The higher number of the used features in PDT agrees with the bigger amount of the training data.

		VALEVAL - data		PDT - dtest		PDT - etest	
Feature type	**#Features**	**#Feat. used**	**Relat. weight**	**#Feat. used**	**Relat. weight**	**#Feat. used**	**Relat. weight**
Morphological	60	27	35.92	44	45.37	44	44.68
Syntactical	103	23	46.28	39	30.76	41	31.59
Idiomatic	118	3	0.85	16	1.20	16	1.20
Animacy	14	8	5.25	9	3.12	9	3.24
WordNet	128	44	11.70	92	19.55	92	19.29
Total	**423**	**105**	**100.00**	**200**	**100.00**	**202**	**100.00**

The column "#Features used" (in %) indicates the number of features used in the decision trees.

The column "Relative weight" indicates the weight based on the feature occurrences in the decision trees.

Table 7.10: Types of features.

Table 7.10 shows the relative weights of the individual feature types for the corresponding corpora. The *relative weight* columns show the relative weights of individual types of features, which was acquired by summing the weights of all used features of the respective types as computed in the previous chapter. We used the weights as described in Section 7.6.2.

The syntax-based features were used most often in the VALEVAL, while morphological features dominated in PDT.

The WordNet features were used relatively often, considering their low accuracies, but they are also the most numerous features.

The idiomatic features had hardly positive weight, as very small number of them were used in the decision trees. We see two main reasons for this fail. First, the idiomatic expression are well discriminative, however, they do not occur often in natural language. Second, we used quite limited idiomatic lexicon (containing only 118 idiomatic expression).

The animacy features scored badly, what can be attributed to the low coverage of the automatic animacy detection, as described in Section 6.5.

7.6.4 Accuracy vs. Precision

Classification methods are usually evaluated using two metrics, the precision and the recall. Nowadays, it is even difficult to imagine a conference paper in the field which is not evaluated using the precision and the recall or in a comparable way.

The **precision** (P) states the portion of the correctly classified instances among all the classified instances:

$$P = \frac{\#\text{correctly assigned instances}}{\#\text{assigned instances}}$$

The **recall** (R) states the portion of the correctly classified instances among all the relevant instances:

$$R = \frac{\#\text{correctly assigned instances}}{\#\text{all relevant instances}}$$

Moreover, we will use the term **coverage** for the portion of the (even incorrectly) classified instances from all the relevant instances.

In the cases where all the instances are classified and only one output is assigned to each, the precision is equal to the recall and this value is usually referred to as **accuracy** (A).

So far, we have always indicated the accuracies in this work. Let us now state a question, whether this indication is correct. The accuracy is the same as the precision under the two already mentioned assumptions: first, we assign only one result to each sample, second, we classify all the samples. As the first assumption clearly holds, we now examine closely the second one.

First we formulate our task:

Assign a verb frame to all autogrammatical[5] *verbs in the corpus for which we have a classifier.*

From this perspective, we are using the term *accuracy* correctly. It can be, however, argued that having or missing classifier is dependent on our implementation and it should not be reflected in the formulation of the task. From this point of view, we should reformulate the task as follows:

Assign a verb frame to all autogrammatical verbs in the corpus (regardless of having a classifier or not).

A new problem arrises – how to classify verbs for which we do not have any classifier because we did not see them in the training data. As we have "no experience" with the verbs, we can only randomly pick up a label from

[5] Autogrammatical verbs correspond to the verbal nodes in the tectogrammatical representation of sentence.

the lexicon[6]. This is closer to *speculation* than *determination*. The other possibility is not to determine the label which leads to equivalent result (classification) as in the original specification of the task. However, what is not equivalent is the evaluation as we have not classified all the verbs which were to classify and therefore $\#assigned\ instances < \#all\ relevant\ instances$, ergo $P > R$.

If we consider the second formulation of the task as more natural we find out that what we called accuracy so far is, in fact, precision and a logical question arrises – what is the recall?

Let us consider the PDT with C5 decision trees classifier using the full feature set for the rest of this section.

The number of assigned frames was 8,711.

The dtest data set contained 1,874 different verbs (etest contained 1,940). Out of this, 1,636 (1,712 for etest) was present in the training dataset as well. No classifier was trained for the 238 remaining verbs (228 for etest).

The number of running verbs in the dtest was 8,970 (9,630 for etest). Out of this, 8,711 was assigned by the disambiguation method (9,381 for etest) because there was a classifier for them. 6,852 of those were assigned correctly (7,309 for etest).

The coverage was relatively high for both datasets – 97.11% for the dtest and 97.41 % for the etest. This is mainly because the training set is 8-times larger than each testing set.

The precision was 78.66% and 77.91% and the recall 76.39% and 75.90%, for the dtest and etest respectively.

The complete statistics is summarized in Table 7.11 (the value *Accuracy* will be explained in the following text).

It is important to realize, that the reasons of the losses in the recall are due to the verbs with low frequencies in the running text. If they did not occur in the training part of the corpus (80 % of the data), we can expect their total amount in the corpus to be low as well[7]. There verbs are usually not very interesting from the linguistical point of view and they have little number of different senses in the lexicon (usually only one).

If we only randomly selected a sense for these verbs from the lexicon, i.e.

[6] There are actually more sophisticated ways than the random selection – e.g. comparing the context of the verb with the lexicon glosses but we leave those methods aside for now.

[7] In fact, their total amount is usually not one, as we could expect from statistical intuition, because they tend to reoccur in the same documents. In this case, they are usually used in the same sense.

	dtest	etest
Verbs in the testing set	1,874	1,940
- also present in the training set	1,636	1,712
- missing in the training set	238	228
Running verbs in the testing set	8,970	9,630
- also present in the training set	8,711	9,381
- missing in the training set	259	249
- correctly assigned	6,852	7,309
Coverage	97.11%	97.41 %
Precision	78.66%	77.91%
Recall	76.39%	75.90%
Accuracy	78.99%	78.23%

All evaluation results are for the C5 decision tree classifier using the full feature set on the Prague Dependency Treebank.
The accuracies are interpolated from the lexicon for verbs which are missing in the training data.

Table 7.11: Accuracy vs. precision/recall

set the accuracy for these verbs to $\frac{1}{\#\text{senses in the lexicon}}$, we would determine a label for each verb in the corpus and we could use the term accuracy again. Using this technique, we gain the accuracy of 78.99% for the dtest and 78.99% for the etest, which is even higher than the original accuracy. This was the answers to the question: *"what result did we achieved for the all-words task"*?

We have shown that if we widen the evaluation model by mindless guessing of the labels for unknown verbs, the overall accuracy increases. At first glance this conclusion might seem controversial.

The next question is what to do if we are about to analyze a verb which is not in the lexicon[8]. We cannot assign any verb frames to such verbs because nothing like this is defined. The solution to assume that there is only one (default) sense is disputable, it might even lead to artificial boosting performance of the method by using a limited lexicon. The only possible solution is to leave the accuracy and return to the precision / recall evaluation again.

[8] For PDT-VALLEX these are all verbs that did not occur in the tectogrammatically annotated part of the Prague Dependency Treebank. Because PDT is a corpus composed of data from a stylistically narrow area (newspaper articles), these verbs might include even quite common, and often ambiguous verbs, like *načíst*, *ožrat*, *červenat* and others.

It is still to mention that such disputation is not relevant for the VALEVAL corpus, as this corpus does not aspire to cover an essential part of the Czech (running) verbs, but as a matter of principle it focuses only on selected verbs. The testing dataset therefore contains only the sentences with the verbs from the selected subpart of the VALLEX lexicon. These sentences contain other verbs as well, but for them the classification in unknown. If we wanted to evaluate the results from the perspective of usability for classification of running text, we would have to take into account the frequencies of the verbs from such text. We also could approximate them by the relative frequencies in the Czech National Corpus. However, VALEVAL was not constructed to cover the language (common running text), therefore it is hardly appropriate to blame it for the low recall.

7.6.5 Ignoring rare verbs

The final results include weighed accuracies of all base lemmas occurring at least once in the training data. However, base-lemmas which occurred only few times do not provide enough samples for the methods to train reliably. This might result in low accuracy (i.e. precision) in the overall result. In this section we check this hypothesis.

We tried to leave out the base lemmas with low frequencies in the training data as we suppose that we can not train the corresponding classifiers properly. As we gained classifiers for some of the original set of lemmas only, we could also classify a smaller part of the testing set, and therefore the recall droped. However, we hoped that the precision would rise because the classification was more reliable.

For a given threshold t we trained the classifiers for lemmas which occurred in the training data at least t-times. We evaluated those classifiers on the testing data, leaving unclassified the verbs whose classifier was not constructed. The relative portion of the classified samples gave the coverage value. The bigger the value of t, the smaller the coverage.

We performed the experiment on the development and the evaluation test set of the Prague Dependency Treebank. Because in the VALEVAL corpus, all base lemmas have the same number of samples, there is no point in doing such experiment.

Tables 7.12 and 7.13 give the values of the precision, recall, and coverage for threshold zero through seven for the development and evaluation test set, respectively.

Threshold	Precision	Recall	Coverage
0	78.66	76.39	97.11
1	78.52	74.83	95.30
2	78.55	73.56	93.65
3	78.40	72.05	91.91
4	78.31	70.95	90.60
5	78.27	69.96	89.38
6	78.32	69.10	88.23
7	78.19	68.07	87.06

Table 7.12: Dependency of the precision, recall and coverage [%] on the threshold of the minimal training sample size - PDT dtest.

Threshold	Precision	Recall	Coverage
0	77.91	75.90	97.41
1	77.70	74.26	95.57
2	77.82	73.18	94.04
3	77.74	71.69	92.22
4	77.77	70.51	90.66
5	77.72	69.43	89.34
6	77.67	68.48	88.17
7	77.64	67.79	87.31

Table 7.13: Dependency of the precision, recall and coverage [%] on the threshold of the minimal training sample size - PDT etest.

The results show that we can not validate the hypothesis proposed in the beginning of this section. All the quantities (the precision, the recall, and the coverage) were falling with the rising threshold. This can be due to the fact that the lemmas with low frequencies have usually less frames (i.e. senses) than the more common verbs, and therefore we were ignoring relatively easy cases while leaving the complicated ones.

7.6.6 Verbs být and mít

Verbs *být* (*to be*) and *mít* (*to have*) have a special position in the PDT. The valency of those two most common Czech verbs is not resolved properly in the corpus, according to authors' statement.

The verb *být* with 46 different frames in the training set and 54 different frames in the PDT-VALLEX has 7,650 out of 9,967 (76.75%) samples annotated to the most common frame (copula *být* – part of a verbnominal predicate). This valency frame often corresponds to very distinct usages of the verb.

On the other hand, the verb *mít* with 78 different frames in the training set and 91 different frames in the PDT-VALLEX[9] has only 488 out of 2,274 (21.46 %) of senses annotated to the most common sense (described by examples *mít pravdu*; *mít zvuk*; *mít ponětí, potuchu*; *mít aférku*; *mít dost práce*; *mít svátek*; *mít premiéru*; *mít vystoupení*; *mít koncert*; *mít pohřeb*), different frames of the verb are very steadily distributed, compared to the corpus average. There are 27 frames for the verb *mít* used only once in the training data which gives the classifier little possibility to train accurately. This suggests that different verb frames are not merged properly, and reprocessing of the lemma would be appropriate in next versions of the lexicon.

Table 7.14 shows the baseline and the accuracy values of the C5 decision trees and the Maximum Entropy classifier, when disregarding the verbs *být* and *mít*, respectively.

	dtest		
	Baseline	C5-DT Accuracy	MaxEnt Accuracy
All verbs	73.19	78.66	79.67
All verbs - *být*	72.53	77.49	78.35
All verbs - *mít*	75.05	80.26	81.00
All verbs - *být* - *mít*	74.72	79.36	79.88
	etest		
	Baseline	C5-DT Accuracy	MaxEnt Accuracy
All verbs	71.98	77.91	78.26
All verbs - *být*	71.24	76.72	76.49
All verbs - *mít*	73.93	79.55	79.80
All verbs - *být* - *mít*	73.54	78.63	78.25

Table 7.14: The baseline and the accuracy values for C5 decision trees and maximum entropy classifier when disregarding the verbs *být* and *mít*

7.6.7 Morphological Context

In this section, we analyze the influence of the width of the morphological context to the performance of the disambiguation in both corpora. So far, we

[9] Counting also verbs with reflexive particles

Diameter	Window size	C5-DT	MaxEnt
0	1	62.60	59.47
1	3	62.21	65.94
2	5	67.62	61.65
3	7	67.56	63.48
4	9	66.05	66.63
5	11	66.05	66.63

Table 7.15: Influence of the size of the morphological context for the VALEVAL corpus.

used the morphological features generated from a five-token window centered around the verb. By comparing different sizes of the morphological window, we show that the size of five tokens is a reasonable choice.

In the following text, we will use the term **diameter** (d) for the number of tokens in the window following and preceeding the verb.

$$\underbrace{\square\ldots\square}_{d}\,verb\,\underbrace{\square\ldots\square}_{d}$$

Therefore, the size of the window is equal to $2d+1$; the five-tokens window has the diameter of two. We have tried the C5 decision trees and the maximum entropy classifier on both corpora using morphological features with the window diameter ranging from zero (one token) to five (eleven tokens).

The morphological features only have the scope within one sentence, if the window exceeds the sentence boundary on either of the sides, the values of corresponding features were assigned a special value (undefined).

Table 7.15 shows the performance of the methods on the VALEVAL corpus. It can be seen that for the C5 method, the window with diameter equal to two performed best. A smaller window does not provide enough information and a larger window confuses the method by adding noise. In the case of the maximum entropy classifier, the morphological window with diameter four scored best, however the model tended to produce more noise in the output. This is mainly because both the input data and the number of features is small, what is not suitable for this classifier.

Table 7.16 shows the same statistics for the Prague Dependency Treebank. Due to larger data sets, the results for the PDT tend to be more balanced than the results for the VALEVAL corpus. Concerning the C5 classifier, the best result was achieved with the morphological window with diameter one. The window with diameter two scored as the second best. Widening

Diameter	Window size	C5-DT	MaxEnt
0	1	73.85	73.78
1	3	75.89	76.39
2	5	75.73	76.58
3	7	75.57	76.52
4	9	75.19	76.49
5	11	75.19	76.49

Table 7.16: Influence of the size of the morphological context for the PDT.

the diameter was confusing the method. The maximum entropy classifier is much more robust in the sense that adding useless features does not hurt the performance of the model, so the confusion is not apparent. However, widening the window does not add any performance progress.

To conclude, when we are looking only at the sentence as a sequence of words, ignoring the syntactical structure, the information about morphology helps, but only from the five tokens surrounding the disambiguated verb. The further the token is, the weaker and more unpredictable is its relation to the verb. Here, the syntactical information can add much more precise knowledge about the relation of the the verb and other lemmas in the sentence.

Chapter 8

Conclusion

The disambiguation of verb senses in Czech has been extensively studied in this thesis. Different machine learning methods and different approaches to WSD and related tasks were introduced.

We investigated which type of information is important to consider when determining the sense of verbs. In fact, instead of senses we used the valency frames. Each verb occurrence was described by hundreds of features of five basic types. The types of the features were evaluated separately and also compared to each other. The most important features turned out to be the ones using information about the surface syntax.

Experiments using different machine learning methods were performed, including the Naïve Bayes Classifier, decision trees, rule-based methods, Maximum Entropy model, and Support Vector Machines. The methods were validated on two qualitatively and quantitatively different corpora — the VALEVAL corpus and the Prague Dependency Treebank. For the smaller VALEVAL corpus, the C5 decision trees and rule-based methods turned out to be the most accurate. For the large Prague Dependency Treebank, the support vector machines and maximum entropy model performed better than other methods.

On the VALEVAL corpus, we achieved improvement 12% absolute over the baseline. On the more challenging Prague Dependency Treebank, improvement 6.5% absolute over the baseline was measured on both the development and the evaluation testing set.

In the evaluation section we investigated the results from different perspectives giving alternative analysis and evaluations.

To summarize the thesis, different techniques of disambiguation of verb senses were proposed, implemented and thoroughly evaluated on two Czech corpora. The achieved improvement over baseline validated the correctness of the underlying ideas.

Further perspectives. Even though this work deals with the disambiguation task, extensively discussing many alternatives, there still remain several directions for the potential extension of the work.

In our opinion, more attention given to the tuning of parameters of non-linear SVM kernels might bring some improvement in performance.

The problem with low number of training samples can be partially avoided by merging aspectual counterparts which often share the valency behavior. However, this might not be applicable for all verbs, and it would require further exploration. We would also need the mapping of aspectual pairs which is part of the VALLEX lexicon but is missing in the PDT-VALLEX.

The proposed methods might also be further adapted to other languages. However, for languages with limited morphology, e.g. English, a revision of features should be considered, as the current feature set is heavily based on information resulting from morphology.

Appendix A

List of Features

A.1 Morphological features

Name	Description
M_-2_1	Part of speech of the word two possitions before the verb
M_-2_2	Detailed part of speech of the word two possitions before the verb
M_-2_3	Gender of the word two possitions before the verb
M_-2_4	Number of the word two possitions before the verb
M_-2_5	Case of the word two possitions before the verb
M_-2_6	Possessor's gender of the word two possitions before the verb
M_-2_7	Possessor's number of the word two possitions before the verb
M_-2_8	Person of the word two possitions before the verb
M_-2_9	Tense of the word two possitions before the verb
M_-2_10	Degree of comparison of the word two possitions before the verb
M_-2_11	Negation of the word two possitions before the verb
M_-2_12	Voice of the word two possitions before the verb
M_-1_1	Part of speech of the word preceeding the verb
M_-1_2	Detailed part of speech of the word preceeding the verb
M_-1_3	Gender of the word preceeding the verb
M_-1_4	Number of the word preceeding the verb
M_-1_5	Case of the word preceeding the verb
M_-1_6	Possessor's gender of the word preceeding the verb
M_-1_7	Possessor's number of the word preceeding the verb
M_-1_8	Person of the word preceeding the verb
M_-1_9	Tense of the word preceeding the verb
M_-1_10	Degree of comparison of the word preceeding the verb
M_-1_11	Negation of the word preceeding the verb
M_-1_12	Voice of the word preceeding the verb
M_0_1	Part of speech of the verb
M_0_2	Detailed part of speech of the verb
M_0_3	Gender of the verb
M_0_4	Number of the verb
M_0_5	Case of the verb
M_0_6	Possessor's gender of the verb
M_0_7	Possessor's number of the verb
M_0_8	Person of the verb
M_0_9	Tense of the verb

Name	Description
M_0_10	Degree of comparison of the verb
M_0_11	Negation of the verb
M_0_12	Voice of the verb
M_1_1	Part of speech of the word following the verb
M_1_2	Detailed part of speech of the word following the verb
M_1_3	Gender of the word following the verb
M_1_4	Number of the word following the verb
M_1_5	Case of the word following the verb
M_1_6	Possessor's gender of the word following the verb
M_1_7	Possessor's number of the word following the verb
M_1_8	Person of the word following the verb
M_1_9	Tense of the word following the verb
M_1_10	Degree of comparison of the word following the verb
M_1_11	Negation of the word following the verb
M_1_12	Voice of the word following the verb
M_2_1	Part of speech of the word two possitions after the verb
M_2_2	Detailed part of speech of the word two possitions after the verb
M_2_3	Gender of the word two possitions after the verb
M_2_4	Number of the word two possitions after the verb
M_2_5	Case of the word two possitions after the verb
M_2_6	Possessor's gender of the word two possitions after the verb
M_2_7	Possessor's number of the word two possitions after the verb
M_2_8	Person of the word two possitions after the verb
M_2_9	Tense of the word two possitions after the verb
M_2_10	Degree of comparison of the word two possitions after the verb
M_2_11	Negation of the word two possitions after the verb
M_2_12	Voice of the word two possitions after the verb

A.2 Syntax-based features

Name	Description
S2_part_se	Presence of reflexive particle *se* dependent on the verb
S2_part_si	Presence of reflexive particle *si* dependent on the verb
S2_inf_verb	Presence of a verb (in infinitive) dependent on the verb
S2_super_verb	The examined verb is dependent on another verb (finite or infinite)
S2_N1	Pres. of noun or a subst. pron. in nominative dep. on the verb
S2_N2	Pres. of noun or a subst. pron. in genitive dep. on the verb
S2_N3	Pres. of noun or a subst. pron. in dative dep. on the verb
S2_N4	Pres. of noun or a subst. pron. in accusative dep. on the verb
S2_N5	Pres. of noun or a subst. pron. in vocative dep. on the verb
S2_N6	Pres. of noun or a subst. pron. in local dep. on the verb
S2_N7	Pres. of noun or a subst. pron. in instrumental dep. on the verb
S2_A1	Pres. of adjective or a adject. pron. in nomin. dep. on the verb
S2_A2	Pres. of adjective or a adject. pron. in genitive dep. on the verb
S2_A3	Pres. of adjective or a adject. pron. in dative dep. on the verb
S2_A4	Pres. of adjective or a adject. pron. in accus. dep. on the verb
S2_A5	Pres. of adjective or a adject. pron. in vocative dep. on the verb

Name	**Description**
S2_A6	Pres. of adjective or a adject. pron. in local dep. on the verb
S2_A7	Pres. of adjective or a adject. pron. in instr. dep. on the verb
S2_prep+1	Presence of preposition in nominative dependent on the verb
S2_prep+2	Presence of preposition in genitive dependent on the verb
S2_prep+3	Presence of preposition in dative dependent on the verb
S2_prep+4	Presence of preposition in accusative dependent on the verb
S2_prep+5	Presence of preposition in vocative dependent on the verb
S2_prep+6	Presence of preposition in local dependent on the verb
S2_prep+7	Presence of preposition in instrumental dependent on the verb
S2_conj_ac3	Presence of subordinate conjunction *ač* dependent on the verb
S2_conj_ac3koli	Presence of subordinate conjunction *ačkoli* dependent on the verb
S2_conj_aby	Presence of subordinate conjunction *aby* dependent on the verb
S2_conj_at3	Presence of subordinate conjunction *ať* dependent on the verb
S2_conj_az3	Presence of subordinate conjunction *až* dependent on the verb
S2_conj_aniz3	Presence of subordinate conjunction *aniž* dependent on the verb
S2_conj_byt3	Presence of subordinate conjunction *byť* dependent on the verb
S2_conj_co	Presence of subordinate conjunction *co* dependent on the verb
S2_conj_coby	Presence of subordinate conjunction *coby* dependent on the verb
S2_conj_jakmile	Presence of subordinate conjunction *jakmile* dependent on the verb
S2_conj_jako	Presence of subordinate conjunction *jako* dependent on the verb
S2_conj_jakoby	Presence of subordinate conjunction *jakoby* dependent on the verb
S2_conj_jakoz3	Presence of subordinate conjunction *jakož* dependent on the verb
S2_conj_jakoz3to	Presence of subordinate conjunction *jakožto* dependent on the verb
S2_conj_jelikoz3	Presence of subordinate conjunction *jelikož* dependent on the verb
S2_conj_jestli	Presence of subordinate conjunction *jestli* dependent on the verb
S2_conj_jestliz3e	Presence of subordinate conjunction *jestliže* dependent on the verb
S2_conj_kdyby	Presence of subordinate conjunction *kdyby* dependent on the verb
S2_conj_kdyz3	Presence of subordinate conjunction *když* dependent on the verb
S2_conj_lec3	Presence of subordinate conjunction *leč* dependent on the verb
S2_conj_liz3	Presence of subordinate conjunction *liž* dependent on the verb
S2_conj_z3e	Presence of subordinate conjunction *že* dependent on the verb
S2_conj_nez3	Presence of subordinate conjunction *než* dependent on the verb
S2_conj_nez3li	Presence of subordinate conjunction *nežli* dependent on the verb
S2_conj_pakliz3e	Presence of subordinate conjunction *pakliže* dependent on the verb
S2_conj_pr3estoz3e	Presence of subordinate conjunction *přestože* dependent on the verb
S2_conj_pr3ic3emz3	Presence of subordinate conjunction *přičemž* dependent on the verb
S2_conj_pokud	Presence of subordinate conjunction *pokud* dependent on the verb
S2_conj_pone3vadz3	Presence of subordinate conjunction *poněvadž* dep. on the verb
S2_conj_protoz3e	Presence of subordinate conjunction *protože* dependent on the verb
S2_conj_sec3	Presence of subordinate conjunction *seč* dependent on the verb
S2_conj_takz3e	Presence of subordinate conjunction *takže* dependent on the verb
S2_conj_tr3ebaz3e	Presence of subordinate conjunction *třebaže* dependent on the verb
S2_conj_zati2mco	Presence of subordinate conjunction *zatímco* dependent on the verb
S2_conj_zato	Presence of subordinate conjunction *zato* dependent on the verb
S2_conj_zda	Presence of subordinate conjunction *zda* dependent on the verb
S2_conj_zdali	Presence of subordinate conjunction *zdali* dependent on the verb
S2_be3hem-2	Presence of preposition *během* in genitive dependent on the verb
S2_bez-2	Presence of preposition *bez* in genitive dependent on the verb

Name	Description
S2_bli2zko-2	Presence of preposition *blízko* in genitive dependent on the verb
S2_di2ky-3	Presence of preposition *díky* in dative dependent on the verb
S2_di2k-3	Presence of preposition *dík* in dative dependent on the verb
S2_dle-2	Presence of preposition *dle* in genitive dependent on the verb
S2_do-2	Presence of preposition *do* in genitive dependent on the verb
S2_kolem-2	Presence of preposition *kolem* in genitive dependent on the verb
S2_kontra-1	Presence of preposition *kontra* in nominative dependent on the verb
S2_krome3-2	Presence of preposition *kromě* in genitive dependent on the verb
S2_k-3	Presence of preposition *k* in dative dependent on the verb
S2_kvu3li-3	Presence of preposition *kvůli* in dative dependent on the verb
S2_mezi-7	Presence of preposition *mezi* in instrumental dependent on the verb
S2_mezi-4	Presence of preposition *mezi* in acousative dependent on the verb
S2_mimo-2	Presence of preposition *mimo* in genitive dependent on the verb
S2_mimo-4	Presence of preposition *mimo* in acousative dependent on the verb
S2_mi2sto-2	Presence of preposition *místo* in genitive dependent on the verb
S2_nad-7	Presence of preposition *nad* in instrumental dependent on the verb
S2_nad-4	Presence of preposition *nad* in acousative dependent on the verb
S2_nami2sto-2	Presence of preposition *namísto* in genitive dependent on the verb
S2_napospas-3	Presence of preposition *napospas* in dative dependent on the verb
S2_naproti-3	Presence of preposition *naproti* in dative dependent on the verb
S2_na-4	Presence of preposition *na* in acousative dependent on the verb
S2_na-6	Presence of preposition *na* in local dependent on the verb
S2_ob-4	Presence of preposition *ob* in acousative dependent on the verb
S2_od-2	Presence of preposition *od* in genitive dependent on the verb
S2_ohledne3-2	Presence of preposition *ohledně* in genitive dependent on the verb
S2_okolo-2	Presence of preposition *okolo* in genitive dependent on the verb
S2_oproti-3	Presence of preposition *oproti* in dative dependent on the verb
S2_o-4	Presence of preposition *o* in acousative dependent on the verb
S2_o-6	Presence of preposition *o* in local dependent on the verb
S2_pobli2z3-2	Presence of preposition *poblíž* in genitive dependent on the verb
S2_pode2l-2	Presence of preposition *podél* in genitive dependent on the verb
S2_podle-2	Presence of preposition *podle* in genitive dependent on the verb
S2_pod-4	Presence of preposition *pod* in acousative dependent on the verb
S2_pod-7	Presence of preposition *pod* in instrumental dependent on the verb
S2_pomoci2-2	Presence of preposition *pomocí* in genitive dependent on the verb
S2_po-4	Presence of preposition *po* in acousative dependent on the verb
S2_prostr3ednictvi2m-2	Presence of preposition *prostřednictvím* in genitive dep. on the verb
S2_proti-3	Presence of preposition *proti* in dative dependent on the verb
S2_pro-4	Presence of preposition *pro* in acousative dependent on the verb
S2_pr3ed-7	Presence of preposition *před* in instrumental dependent on the verb
S2_pr3ed-4	Presence of preposition *před* in acousative dependent on the verb
S2_pr3es-4	Presence of preposition *přes* in acousative dependent on the verb
S2_pr3i-6	Presence of preposition *při* in local dependent on the verb
S2_skrz-4	Presence of preposition *skrz* in acousative dependent on the verb
S2_stran-2	Presence of preposition *stran* in genitive dependent on the verb
S2_s-7	Presence of preposition *s* in instrumental dependent on the verb
S2_s-2	Presence of preposition *s* in genitive dependent on the verb
S2_uprostr3ed-2	Presence of preposition *uprostřed* in genitive dependent on the verb

Name	Description
S2_u-2	Presence of preposition *u* in genitive dependent on the verb
S2_uvnitr3-2	Presence of preposition *uvnitř* in genitive dependent on the verb
S2_vc3etne3-2	Presence of preposition *včetně* in genitive dependent on the verb
S2_vedle-2	Presence of preposition *vedle* in genitive dependent on the verb
S2_versus-1	Presence of preposition *versus* in nominative dependent on the verb
S2_vne3-2	Presence of preposition *vně* in genitive dependent on the verb
S2_vstr3i2c-3	Presence of preposition *vstříc* in dative dependent on the verb
S2_v-4	Presence of preposition *v* in acousative dependent on the verb
S2_v-6	Presence of preposition *v* in local dependent on the verb
S2_vu3c3i-3	Presence of preposition *vůči* in dative dependent on the verb
S2_vu2kol-2	Presence of preposition *vúkol* in genitive dependent on the verb
S2_vyjma-2	Presence of preposition *vyjma* in genitive dependent on the verb
S2_vzdor-3	Presence of preposition *vzdor* in dative dependent on the verb
S2_za-2	Presence of preposition *za* in genitive dependent on the verb
S2_za-4	Presence of preposition *za* in accusative dependent on the verb
S2_za-7	Presence of preposition *za* in instrumental dependent on the verb
S2_zpod-2	Presence of preposition *zpod* in genitive dependent on the verb
S2_zpoza-2	Presence of preposition *zpoza* in genitive dependent on the verb
S2_z-2	Presence of preposition *za* in genitive dependent on the verb

A.3 Anymacy features

Name	Description
A_anym1	Presence of an animate substantive in nominative
A_anym2	Presence of an animate substantive in genitive
A_anym3	Presence of an animate substantive in dative
A_anym4	Presence of an animate substantive in acusative
A_anym5	Presence of an animate substantive in vocativ
A_anym6	Presence of an animate substantive in local
A_anym7	Presence of an animate substantive in instrumental
A_V_anym1	Presence of an animate substantive in nominative dependent on the verb
A_V_anym2	Presence of an animate substantive in genitive dependent on the verb
A_V_anym3	Presence of an animate substantive in dative dependent on the verb
A_V_anym4	Presence of an animate substantive in acusative dependent on the verb
A_V_anym5	Presence of an animate substantive in vocativ dependent on the verb
A_V_anym6	Presence of an animate substantive in local dependent on the verb
A_V_anym7	Presence of an animate substantive in instrumental dependent on the verb

A.4 Idiomatic features

Name	Description
V_u2c3ty	Presence of the idionatic expression *účty* in the sentence
V_u2lohu	Presence of the idionatic expression *úlohu* in the sentence
V_bacha	Presence of the idionatic expression *bacha* in the sentence
V_barvu	Presence of the idionatic expression *barvu* in the sentence

Name	**Description**
V_co_mluvit	Presence of the idionatic expression *co mluvit* in the sentence
V_du3lez3itost	Presence of the idionatic expression *důležitost* in the sentence
V_do_de3jin	Presence of the idionatic expression *do dějin* in the sentence
V_do_dus3e	Presence of the idionatic expression *do duše* in the sentence
V_do_formy	Presence of the idionatic expression *do formy* in the sentence
V_do_gala	Presence of the idionatic expression *do gala* in the sentence
V_do_hlavy	Presence of the idionatic expression *do hlavy* in the sentence
V_dohromady	Presence of the idionatic expression *dohromady* in the sentence
V_do_chodu	Presence of the idionatic expression *do chodu* in the sentence
V_do_jine2ho_stavu	Presence of the idionatic expression *do jiného stavu* in the sentence
V_do_karet	Presence of the idionatic expression *do karet* in the sentence
V_do_kr3i2z3ku	Presence of the idionatic expression *do křížku* in the sentence
V_do_kroku	Presence of the idionatic expression *do kroku* in the sentence
V_do_z3aludku	Presence of the idionatic expression *do žaludku* in the sentence
V_do_nebe	Presence of the idionatic expression *do nebe* in the sentence
V_do_noty	Presence of the idionatic expression *do noty* in the sentence
V_do_ochrany	Presence of the idionatic expression *do ochrany* in the sentence
V_do_oka	Presence of the idionatic expression *do oka* in the sentence
V_do_rukou	Presence of the idionatic expression *do rukou* in the sentence
V_do_ruky	Presence of the idionatic expression *do ruky* in the sentence
V_do_situace	Presence of the idionatic expression *do situace* in the sentence
V_do_tempa	Presence of the idionatic expression *do tempa* in the sentence
V_do_tuhe2ho	Presence of the idionatic expression *do tuhého* in the sentence
V_do_vazby	Presence of the idionatic expression *do vazby* in the sentence
V_do_zapomne3ni2	Presence of the idionatic expression *do zapomnění* in the sentence
V_dver3e	Presence of the idionatic expression *dveře* in the sentence
V_hlas	Presence of the idionatic expression *hlas* in the sentence
V_hlavu	Presence of the idionatic expression *hlavu* in the sentence
V_hru3zu	Presence of the idionatic expression *hrůzu* in the sentence
V_jako_v_bavlnce	Presence of the idionatic expression *jako v bavlnce* in the sentence
V_k_dobru	Presence of the idionatic expression *k dobru* in the sentence
V_ke_zdi	Presence of the idionatic expression *ke zdi* in the sentence
V_k_ledu	Presence of the idionatic expression *k ledu* in the sentence
V_ku3z3i	Presence of the idionatic expression *kůži* in the sentence
V_kolem_krku	Presence of the idionatic expression *kolem krku* in the sentence
V_konce	Presence of the idionatic expression *konce* in the sentence
V_korunu	Presence of the idionatic expression *korunu* in the sentence
V_krok	Presence of the idionatic expression *krok* in the sentence
V_kroky	Presence of the idionatic expression *kroky* in the sentence
V_k_sobe3	Presence of the idionatic expression *k sobě* in the sentence
V_k_te3lu	Presence of the idionatic expression *k tělu* in the sentence
V_z3ilou	Presence of the idionatic expression *žilou* in the sentence
V_z3ivot	Presence of the idionatic expression *život* in the sentence
V_mra2z	Presence of the idionatic expression *mráz* in the sentence
V_nade3je	Presence of the idionatic expression *naděje* in the sentence
V_nade3ji	Presence of the idionatic expression *naději* in the sentence
V_na_hlavu	Presence of the idionatic expression *na hlavu* in the sentence
V_najevo	Presence of the idionatic expression *najevo* in the sentence

Name	**Description**
V_na_krk	Presence of the idionatic expression *na krk* in the sentence
V_na_r3adu	Presence of the idionatic expression *na řadu* in the sentence
V_na_lopatky	Presence of the idionatic expression *na lopatky* in the sentence
V_naz3ivu	Presence of the idionatic expression *naživu* in the sentence
V_na_milost	Presence of the idionatic expression *na milost* in the sentence
V_na_mus3ku	Presence of the idionatic expression *na mušku* in the sentence
V_na_mysli	Presence of the idionatic expression *na mysli* in the sentence
V_na_oc3i2ch	Presence of the idionatic expression *na očích* in the sentence
V_na_pas3ka2l	Presence of the idionatic expression *na paškál* in the sentence
V_na_pr3etr3es	Presence of the idionatic expression *na přetřes* in the sentence
V_napospas	Presence of the idionatic expression *napospas* in the sentence
V_na_povrch	Presence of the idionatic expression *na povrch* in the sentence
V_na_pravou_mi2ru	Presence of the idionatic expression *na pravou míru* in the sentence
V_na_starost	Presence of the idionatic expression *na starost* in the sentence
V_na_sve3tlo	Presence of the idionatic expression *na světlo* in the sentence
V_na_trh	Presence of the idionatic expression *na trh* in the sentence
V_na_ve3domi2	Presence of the idionatic expression *na vědomí* in the sentence
V_na_vlastni2_nohy	Presence of the idionatic expression *na vlastní nohy* in the sentence
V_ohled	Presence of the idionatic expression *ohled* in the sentence
V_pr3ed_oc3ima	Presence of the idionatic expression *před očima* in the sentence
V_pr3ed_rozhodnuti2	Presence of the idionatic expression *před rozhodnutí* in the sentence
V_pr3es_srdce	Presence of the idionatic expression *přes srdce* in the sentence
V_pr3i_z3ivote3	Presence of the idionatic expression *při životě* in the sentence
V_pod_ochranu	Presence of the idionatic expression *pod ochranu* in the sentence
V_pohledem	Presence of the idionatic expression *pohledem* in the sentence
V_pohledy	Presence of the idionatic expression *pohledy* in the sentence
V_po_pra2ci	Presence of the idionatic expression *po práci* in the sentence
V_pozor	Presence of the idionatic expression *pozor* in the sentence
V_pozornost	Presence of the idionatic expression *pozornost* in the sentence
V_pro_a_proti	Presence of the idionatic expression *pro a proti* in the sentence
V_pro_sebe	Presence of the idionatic expression *pro sebe* in the sentence
V_roha	Presence of the idionatic expression *roha* in the sentence
V_roli	Presence of the idionatic expression *roli* in the sentence
V_ruku	Presence of the idionatic expression *ruku* in the sentence
V_sa2m	Presence of the idionatic expression *sám* in the sentence
V_samo_sebou	Presence of the idionatic expression *samo sebou* in the sentence
V_slovo	Presence of the idionatic expression *slovo* in the sentence
V_spolu	Presence of the idionatic expression *spolu* in the sentence
V_s_sebou	Presence of the idionatic expression *s sebou* in the sentence
V_sve3tlo_sve3ta	Presence of the idionatic expression *světlo světa* in the sentence
V_v_u2z3as	Presence of the idionatic expression *v úžas* in the sentence
V_v_u2vahu	Presence of the idionatic expression *v úvahu* in the sentence
V_ve_zna2most	Presence of the idionatic expression *ve známost* in the sentence
V_vhod	Presence of the idionatic expression *vhod* in the sentence
V_v_kolenou	Presence of the idionatic expression *v kolenou* in the sentence
V_vs3ecko	Presence of the idionatic expression *všecko* in the sentence
V_vs3echno	Presence of the idionatic expression *všechno* in the sentence
V_v_oc3i2ch	Presence of the idionatic expression *v očích* in the sentence

Name	Description
V_v_platnost	Presence of the idionatic expression *v platnost* in the sentence
V_v_pove3domi2	Presence of the idionatic expression *v povědomí* in the sentence
V_vstr3i2c	Presence of the idionatic expression *vstříc* in the sentence
V_za2dy	Presence of the idionatic expression *zády* in the sentence
V_za2jem	Presence of the idionatic expression *zájem* in the sentence
V_za2vaz3nost	Presence of the idionatic expression *závažnost* in the sentence
V_za_hlavu	Presence of the idionatic expression *za hlavu* in the sentence
V_za_sebou	Presence of the idionatic expression *za sebou* in the sentence
V_zasve2	Presence of the idionatic expression *zasvé* in the sentence
V_zavde3k	Presence of the idionatic expression *zavděk* in the sentence
V_z_dlane3	Presence of the idionatic expression *z dlaně* in the sentence
V_ze_zr3etele	Presence of the idionatic expression *ze zřetele* in the sentence
V_z_hlavy	Presence of the idionatic expression *z hlavy* in the sentence
V_z_kopy2tka	Presence of the idionatic expression *z kopýtka* in the sentence
V_zr3etel	Presence of the idionatic expression *zřetel* in the sentence
V_z_mysli	Presence of the idionatic expression *z mysli* in the sentence
V_z_oc3i2	Presence of the idionatic expression *z očí* in the sentence
V_zpe3t	Presence of the idionatic expression *zpět* in the sentence

A.5 WerbNet features

Name	Description
W_Agentive	Presence of a noun from semantic class *Agentive* in the sentence
W_Animal	Presence of a noun from semantic class *Animal* in the sentence
W_Artifact	Presence of a noun from semantic class *Artifact* in the sentence
W_BoundedEvent	Presence of a noun from semantic class *BoundedEvent* in the sent.
W_Building	Presence of a noun from semantic class *Building* in the sentence
W_Cause	Presence of a noun from semantic class *Cause* in the sentence
W_Comestible	Presence of a noun from semantic class *Comestible* in the sentence
W_Communication	Presence of a noun from semantic class *Communication* in the sent.
W_Composition	Presence of a noun from semantic class *Composition* in the sentence
W_Condition	Presence of a noun from semantic class *Condition* in the sentence
W_Container	Presence of a noun from semantic class *Container* in the sentence
W_Covering	Presence of a noun from semantic class *.9Covering* in the sentence
W_Creature	Presence of a noun from semantic class *Creature* in the sentence
W_Dynamic	Presence of a noun from semantic class *Dynamic* in the sentence
W_Existence	Presence of a noun from semantic class *Existence* in the sentence
W_Experience	Presence of a noun from semantic class *Experience* in the sentence
W_Form	Presence of a noun from semantic class *Form* in the sentence
W_Function	Presence of a noun from semantic class *Function* in the sentence
W_Furniture	Presence of a noun from semantic class *Furniture* in the sentence
W_Garment	Presence of a noun from semantic class *Garment* in the sentence
W_Gas	Presence of a noun from semantic class *Gas* in the sentence
W_Group	Presence of a noun from semantic class *Group* in the sentence
W_Human	Presence of a noun from semantic class *Human* in the sentence
W_ImageRepresentation	Pres. of a noun from sem. class *ImageRepresentation* in the sent.
W_Instrument	Presence of a noun from semantic class *Instrument* in the sentence

Name	**Description**
W_LanguageRepresentation	Pres. of a noun from sem. class *LanguageRepresentation* in the sent.
W_Liquid	Presence of a noun from semantic class *Liquid* in the sentence
W_Living	Presence of a noun from semantic class *Living* in the sentence
W_Location	Presence of a noun from semantic class *Location* in the sentence
W_Manner	Presence of a noun from semantic class *Manner* in the sentence
W_Mental	Presence of a noun from semantic class *Mental* in the sentence
W_Modal	Presence of a noun from semantic class *Modal* in the sentence
W_MoneyRepresentation	Pres. of a noun from sem. class *MoneyRepresentation* in the sent.
W_Natural	Presence of a noun from semantic class *Natural* in the sentence
W_Object	Presence of a noun from semantic class *Object* in the sentence
W_Occupation	Presence of a noun from semantic class *Occupation* in the sentence
W_Origin	Presence of a noun from semantic class *Origin* in the sentence
W_Part	Presence of a noun from semantic class *Part* in the sentence
W_Phenomenal	Presence of a noun from semantic class *Phenomenal* in the sentence
W_Physical	Presence of a noun from semantic class *Physical* in the sentence
W_Place	Presence of a noun from semantic class *Place* in the sentence
W_Plant	Presence of a noun from semantic class *Plant* in the sentence
W_Possession	Presence of a noun from semantic class *Possession* in the sentence
W_Property	Presence of a noun from semantic class *Property* in the sentence
W_Purpose	Presence of a noun from semantic class *Purpose* in the sentence
W_Quantity	Presence of a noun from semantic class *Quantity* in the sentence
W_Relation	Presence of a noun from semantic class *Relation* in the sentence
W_Representation	Presence of a noun from semantic class *Representation* in the sent.
W_SituationComponent	Presence of a noun from sem. class *SituationComponent* in the sent.
W_SituationType	Presence of a noun from semantic class *SituationType* in the sent.
W_Social	Presence of a noun from semantic class *Social* in the sentence
W_Software	Presence of a noun from semantic class *Software* in the sentence
W_Solid	Presence of a noun from semantic class *Solid* in the sentence
W_Static	Presence of a noun from semantic class *Static* in the sentence
W_Stimulating	Presence of a noun from semantic class *Stimulating* in the sentence
W_Substance	Presence of a noun from semantic class *Substance* in the sentence
W_Time	Presence of a noun from semantic class *Time* in the sentence
W_Top	Presence of a noun from semantic class *Top* in the sentence
W_UnboundedEvent	Presence of a noun from semantic class *UnboundedEvent* in the sent.
W_Usage	Presence of a noun from semantic class *Usage* in the sentence
W_Vehicle	Presence of a noun from semantic class *Vehicle* in the sentence
W_1stOrderEntity	Presence of a noun from semantic class *1stOrderEntity* in the sent.
W_2ndOrderEntity	Presence of a noun from semantic class *2ndOrderEntity* in the sent.
W_3rdOrderEntity	Presence of a noun from semantic class *3rdOrderEntity* in the sent.
W_v_Agentive	Presence of a noun from semantic class *Agentive* dep. on the verb
W_v_Animal	Presence of a noun from semantic class *Animal* dep. on the verb
W_v_Artifact	Presence of a noun from semantic class *Artifact* dep. on the verb
W_v_BoundedEvent	Pres. of a noun from semantic class *BoundedEvent* dep. on the verb
W_v_Building	Presence of a noun from sem. class *Building* dep. on the verb
W_v_Cause	Presence of a noun from semantic class *Cause* dep. on the verb
W_v_Comestible	Presence of a noun from semantic class *Comestible* dep. on the verb
W_v_Communication	Presence of a noun from sem. class *Communication* dep. on the verb
W_v_Composition	Pre. of a noun from semantic class *Composition* dep. on the verb

Name	**Description**
W_v_Condition	Presence of a noun from semantic class *Condition* dep. on the verb
W_v_Container	Presence of a noun from semantic class *Container* dep. on the verb
W_v_Covering	Presence of a noun from semantic class *Covering* dep. on the verb
W_v_Creature	Presence of a noun from semantic class *Creature* dep. on the verb
W_v_Dynamic	Presence of a noun from semantic class *Dynamic* dep. on the verb
W_v_Existence	Presence of a noun from semantic class *Existence* dep. on the verb
W_v_Experience	Presence of a noun from semantic class *Experience* dep. on the verb
W_v_Form	Presence of a noun from semantic class *Form* dep. on the verb
W_v_Function	Presence of a noun from semantic class *Function* dep. on the verb
W_v_Furniture	Presence of a noun from semantic class *Furniture* dep. on the verb
W_v_Garment	Presence of a noun from semantic class *Garment* dep. on the verb
W_v_Gas	Presence of a noun from semantic class *Gas* dep. on the verb
W_v_Group	Presence of a noun from semantic class *Group* dep. on the verb
W_v_Human	Presence of a noun from semantic class *Human* dep. on the verb
W_v_ImageRepresentation	Pres. of a noun from sem. cl. *ImageRepresentation* dep. on the verb
W_v_Instrument	Presence of a noun from sem. cl. *Instrument* dep. on the verb
W_v_LanguageRepresentation	Pres. of a noun from s. c. *LanguageRepresentation* d. on the verb
W_v_Liquid	Presence of a noun from semantic class *Liquid* dep. on the verb
W_v_Living	Presence of a noun from semantic class *Living* dep. on the verb
W_v_Location	Presence of a noun from semantic class *Location* dep. on the verb
W_v_Manner	Presence of a noun from semantic class *Manner* dep. on the verb
W_v_Mental	Presence of a noun from semantic class *Mental* dep. on the verb
W_v_Modal	Presence of a noun from semantic class *Modal* dep. on the verb
W_v_MoneyRepresentation	Pres. of a noun from s. c. *MoneyRepresentation* dep. on the verb
W_v_Natural	Presence of a noun from semantic class *Natural* dep. on the verb
W_v_Object	Presence of a noun from semantic class *Object* dep. on the verb
W_v_Occupation	Presence of a noun from semantic class *Occupation* dep. on the verb
W_v_Origin	Presence of a noun from semantic class *Origin* dep. on the verb
W_v_Part	Presence of a noun from semantic class *Part* dep. on the verb
W_v_Phenomenal	Presence of a noun from sem. class *Phenomenal* dep. on the verb
W_v_Physical	Presence of a noun from semantic class *Physical* dep. on the verb
W_v_Place	Presence of a noun from semantic class *Place* dep. on the verb
W_v_Plant	Presence of a noun from semantic class *Plant* dep. on the verb
W_v_Possession	Presence of a noun from semantic class *Possession* dep. on the verb
W_v_Property	Presence of a noun from semantic class *Property* dep. on the verb
W_v_Purpose	Presence of a noun from semantic class *Purpose* dep. on the verb
W_v_Quantity	Presence of a noun from semantic class *Quantity* dep. on the verb
W_v_Relation	Presence of a noun from semantic class *Relation* dep. on the verb
W_v_Representation	Presence of a noun from sem. cl. *Representation* dep. on the verb
W_v_SituationComponent	Pres. of a noun from sem. cl. *SituationComponent* dep. on the verb
W_v_SituationType	Presence of a noun from sem. cl. *SituationType* dep. on the verb
W_v_Social	Presence of a noun from semantic class *Social* dep. on the verb
W_v_Software	Presence of a noun from semantic class *Software* dep. on the verb
W_v_Solid	Presence of a noun from semantic class *Solid* dep. on the verb
W_v_Static	Presence of a noun from semantic class *Static* dep. on the verb
W_v_Stimulating	Presence of a noun from semantic class *Stimulating* dep. on the verb
W_v_Substance	Presence of a noun from semantic class *Substance* dep. on the verb
W_v_Time	Presence of a noun from semantic class *Time* dep. on the verb

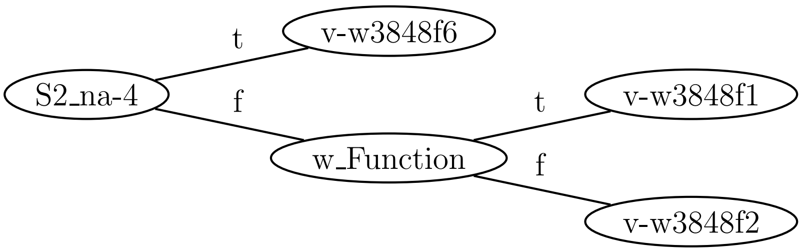
Name	**Description**
W_v_Top	Presence of a noun from semantic class *Top* dep. on the verb
W_v_UnboundedEvent	Presence of a noun from sem. class *UnboundedEvent* dep. on the verb
W_v_Usage	Presence of a noun from semantic class *Usage* dep. on the verb
W_v_Vehicle	Presence of a noun from semantic class *Vehicle* dep. on the verb
W_v_1stOrderEntity	Presence of a noun from sem. class *1stOrderEntity* dep. on the verb
W_v_2ndOrderEntity	Presence of a noun from sem. class *2ndOrderEntity* dep. on the verb
W_v_3rdOrderEntity	Presence of a noun from sem. class *3rdOrderEntity* dep. on the verb

Appendix B

Decision Trees

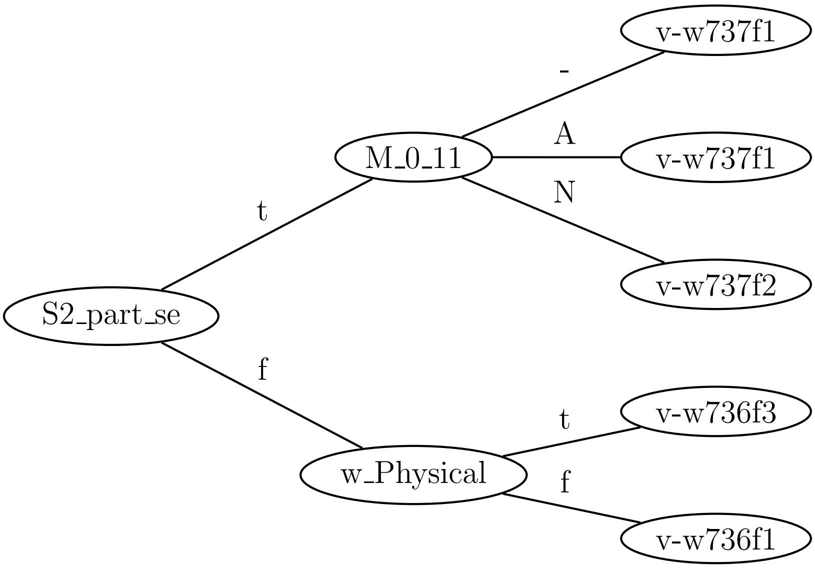
This Appendix shows selected decision trees generated by the C5 algorithm from the training data-set of the PDT.

ponechat



v-w3848f1	nevzít • ponechat jim většinový podíl • p. si naději • p. mamince šátek • p. si právo
v-w3848f2	nechat, zanechat • ponechat knihu na stole
v-w3848f6	 • zemi vraždění napospas

dostávat



v-w736f1

- film dostával od kritiků dobré recenze
- d. kytku od manžela
- za sto tisíc.EXT d. dovolenou u moře
- za otce, místo otce.SUBS d. cenu
- za peníze.MEANS d. v tomto státě všechno
- za zásluhy, za trest.CAUS v-w736f10

v-w736f3

- dostával facky od kamaráda
- d. rány z mnoha stran.DIR1 od mnoha lidí
- d. ránu do hlavy.DIR3

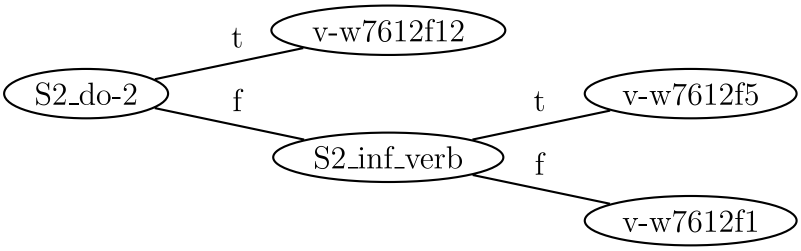
v-w737f1

- dostával se do práce
- d. se mu.BEN do rukou práce studentů

v-w737f2 dostat se

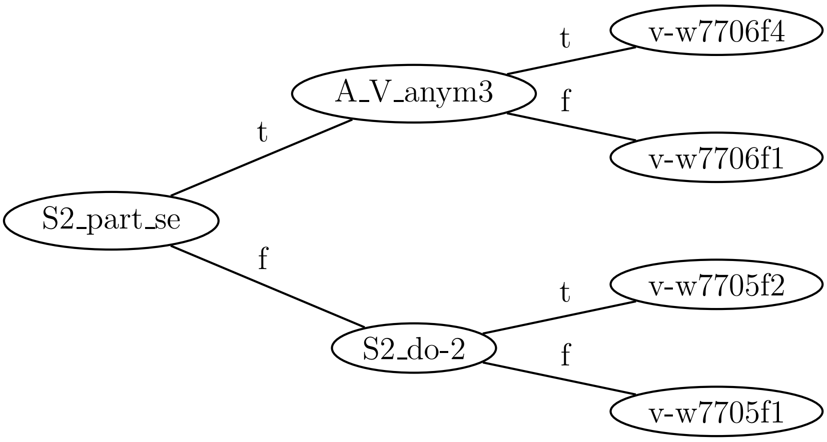
- dostávalo se mu výchovy
- d. se jim zadostiučinění
- nedostávalo se mu odvahy
- d. se jim od mužů vlídného zacházení

vidět



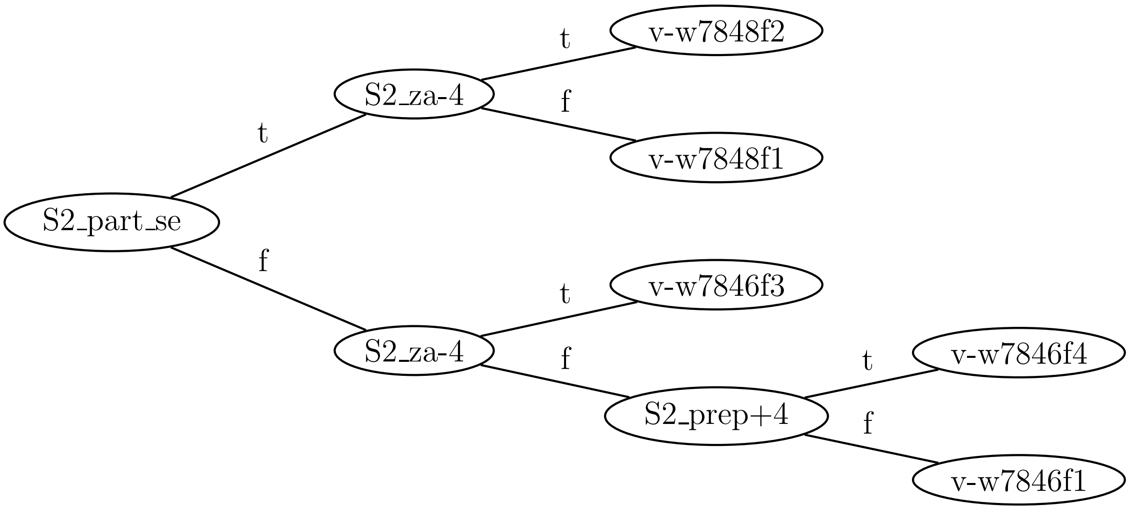
v-w7612f1	uvidět, spatřit • viděl Petra • v., že je unavený • v., kam až to vede • v. ji unavenou.COMPL • neviděl jiné cesty v-w7612f10
v-w7612f12	znát • vidíme do všech řešení • v. nám.BEN do problému
v-w7612f5	 • vidí chlapce přicházet • v. ho, že přichází • v. ji, jak přichází

vrátit



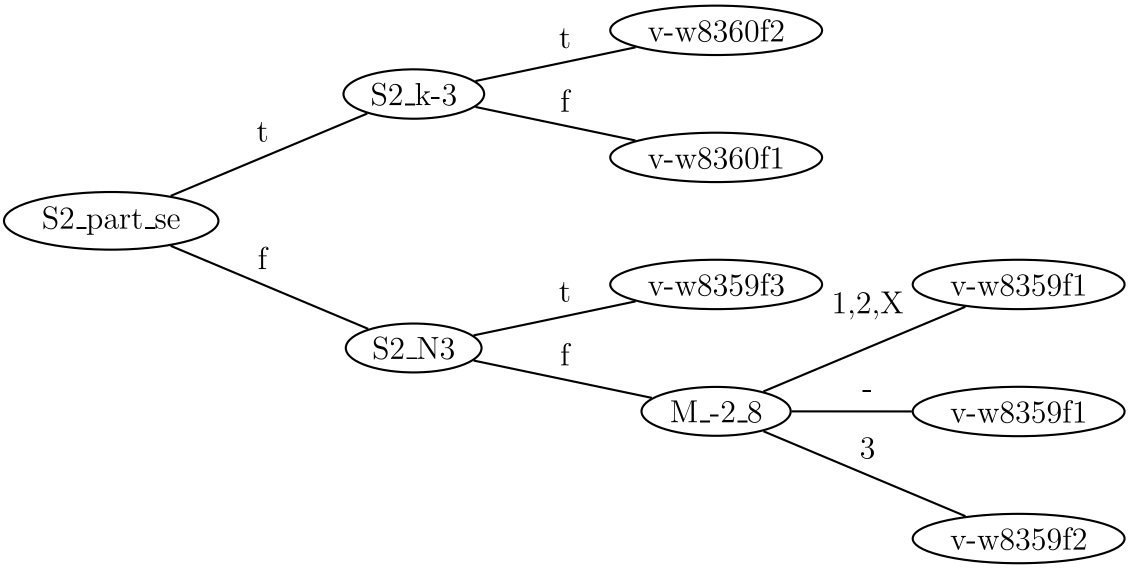
v-w7705f1	navrátit • vrátil mu knihy • do knihovny.DIR3 • v. církvi majetek
v-w7705f2	navrátit • vrátil knihu do knihovny(=knihovně)
v-w7706f1	přijít zpět • vrátit se zpět • v. se do Prahy bez nálady.ACMP
v-w7706f4	obnovit se, začít znovu existovat • nemoc se často vrátila • v. se mu.BEN síly

vydávat



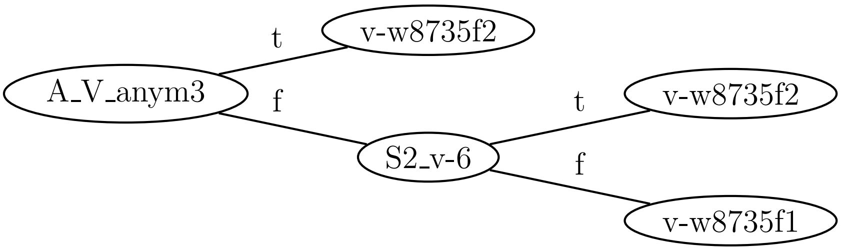
v-w7846f1	publikovat, uveřejnit
	• vydávat knihy
	• v. prohlášení
	• veterinář v. každoročně psovi.BEN osvědčení o vzteklině
	• nakladatel mu.BEN každoročně v. jeho sbírku
v-w7846f3	považovat, prohlašovat
	• vydával neznámého za svého přítele
v-w7846f4	utrácet, platit CO
	• vydávali peníze za získání bytu
	• v. prostředky na získání.AIM jídla
v-w7848f1	jít, jet
	• vydával se do školy
	• v. se za přítelem vlakem.MEANS
v-w7848f2	prohlašovat se, považovat se
	• vydávat se za majora

vyslovit



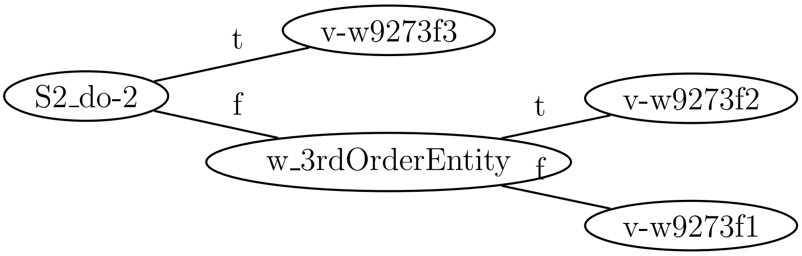
v-w8359f1	vyjádřit, formulovat • vyslovil svůj názor
v-w8359f2	vyhrknout, vyřknout • vyslovit písmenko ř
v-w8359f3	projevit, vyjádřit • vyslovil mu důvěru
v-w8360f1	vyjádřit se • vyslovit se proti extremismu • v. se za sjednocení pravidel • v. se ve prospěch zrušení • cyklista se v. za sjednocení pravidel
v-w8360f2	vyjádřit se • vyslovil se k této otázce • v. se v této záležitosti

zabránit



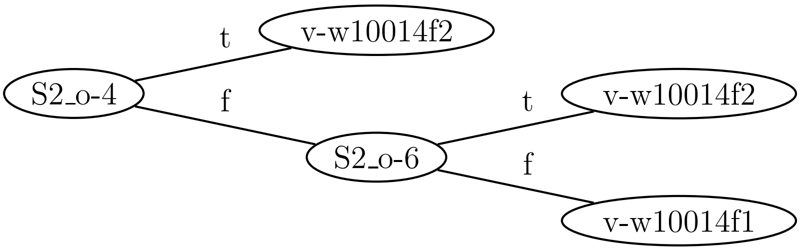
v-w8735f1	odvrátit
	• zabránit provokacím
v-w8735f2	znemožnit
	• zabránil jí, aby odešla
	• nesoustředěnost z. studentům vykonat zkoušku

zavést



v-w9273f1	uvést do chodu, instalovat, zřídit
	• v továrně zavedli píchačky
	• z. nám.BEN delší pracovní dobu
	• z. účet u KB.LOC
	• z. mu.BEN účet
v-w9273f2	nastolit, uplatnit, prosadit
	• zavést stejné podmínky
	• pro všechny.BEN
	• z. nový pořádek
v-w9273f3	odvést, umístit
	• zavedl dítě do školky
	• Petr z. tlupu do lesa

žádat



v-w10014f1	vyžadovat, chtít • žádat od někoho omluvu • ž., aby se omluvil • tato práce po nich ž. zručnost • ž. auto pro manželku.BEN • ž. za výpomoc.CAUS nový byt • ž. za podnájem.SUBS nový byt
v-w10014f2	prosit • žádat někoho, aby se omluvil • ž. nás o omluvu

Bibliography

[Agirre and Edmonds, 2006] Agirre, E. and Edmonds, P. (2006). *Word Sense Disambiguation: Algorithms and Applications (Text, Speech and Language Technology).* Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Aldrich, 1997] Aldrich, J. (1997). R.a. fisher and the making of maximum likelihood 1912-1922, john aldrich. *Statistical Science*, 12(3):162 – 176.

[Baker and Sato, 2003] Baker, C. F. and Sato, H. (2003). The framenet data and software. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 161–164, Morristown, NJ, USA. Association for Computational Linguistics.

[Berger et al., 1996] Berger, A. L., Pietra, S. D., and Pietra, V. J. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

[Bojar et al., 2005] Bojar, O., Semecký, J., and Benešová, V. (2005). VALEVAL: Testing VALLEX Consistency and Experimenting with Word-Frame Disambiguation. *Prague Bulletin of Mathematical Linguistics*, 83:5–17.

[Boser et al., 1992] Boser, B. E., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pages 144–152.

[Breiman et al., 1993] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1993). *Classification and Regression Trees.* Chapman & Hall, New York.

[Buntine, 1990] Buntine, W. (1990). *A Theory of Learning Classification Rules.* PhD thesis, University of Technology, School of Computing Science, Sydney, Sydney.

[Buntine, 1993] Buntine, W. (1993). Learning classification trees. In Hand, D. J., editor, *Artificial Intelligence frontiers in statistics*, pages 182–201. Chapman & Hall,London.

[Charniak, 2000] Charniak, E. (2000). A Maximum-Entropy-Inspired Parser. In *Proceedings of NAACL-2000*, pages pp. 132–139, Seattle, Washington, USA.

[Cikhart and Hajič, 1999] Cikhart, O. and Hajič, J. (1999). Word sense disambiguation of czech texts. In *TSD '99: Proceedings of the Second International Workshop on Text, Speech and Dialogue*, pages 109–114, London, UK. Springer-Verlag.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.*, 20(3):273–297.

[Dang and Palmer, 2005] Dang, H. T. and Palmer, M. (2005). The role of semantic roles in disambiguating verb senses. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 42–49, Morristown, NJ, USA. Association for Computational Linguistics.

[Darroch and Ratcliff, 1972] Darroch, J. and Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *Ann. Math. Statistics*, 43:1470–1480.

[Della Pietra et al., 1997] Della Pietra, S., Della Pietra, V. J., and Lafferty, J. D. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.

[Erk, 2005] Erk, K. (2005). Frame assignment as word sense disambiguation. In *Sixth International Workshop on Computational Semantics (IWCS)*, Tilburg.

[Escudero et al., 2000] Escudero, G., Marquez, L., and Rigau, G. (2000). An empirical study of the domain dependence of supervised word sense disambiguation systems.

[Fellbaum, 1998] Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

[Fillmore, 1976] Fillmore, C. J. (1976). 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*, volume 280, pages 20–32.

[Florian et al., 2002] Florian, R., Cucerzan, S., Schafer, C., and Yarowsky, D. (2002). Combining classifiers for word sense disambiguation. *Natural Language Engineering*, 8(4):327–341.

[Frank and Semecký, 2004] Frank, A. and Semecký, J. (2004). Corpus-based induction of an lfg syntax-semantics interface for frame semantic processing. In Hansen-Schirra, S., Oepen, S., and Uszkoreit, H., editors, *Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora (LINC-04)*, Geneva, Switzerland.

[Hajič, 2000] Hajič, J. (2000). Morphological Tagging: Data vs. Dictionaries. In *Proceedings of the 6th Applied Natural Language Processing and the 1st NAACL Conference*, pages 94–101, Seattle, Washington.

[Hajič, 2004] Hajič, J. (2004). Complex Corpus Annotation: The Prague Dependency Treebank. Bratislava, Slovakia. Jazykovedný ústav Ĺ. Štúra, SAV.

[Hajič and Honetschläger, 2003] Hajič, J. and Honetschläger, V. (2003). Annotation Lexicons: Using the Valency Lexicon for Tectogrammatical Annotation. *Prague Bulletin of Mathematical Linguistics*, (79–80):61–86.

[Hajič et al., 2003] Hajič, J., Panevová, J., Urešová, Z., Bémová, A., Kolářová-Řezníčková‘, V., and Pajas, P. (2003). PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation. In Nivre, J. and Hinrichs, E., editors, *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, pages 57–68, Vaxjo, Sweden. Vaxjo University Press.

[Hajičová, 2002] Hajičová, E. (2002). Theoretical description of language as a basis of corpus annotation: The case of Prague Dependency Treebank. *Prague Linguistic Circle Papers*, 4:111–127.

[Hajič et al., 2006] Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M., Žabokrtský, Z., and Ševčíková Razímová, M. (2006). *Prague Dependency Treebank 2.0.* Linguistic Data Consortium, Philadelphia, PA, USA.

[Hana Skoumalová, 2001] Hana Skoumalová (2001). *Czech syntactic lexicon.* PhD thesis, Charled Univerisy.

[Johnson and Fillmore, 2000] Johnson, C. and Fillmore, C. (2000). The framenet tagset for framesemantic and syntactic coding of predicate-argument structure.

[Kaethe, 2000] Kaethe, E., editor (2000). *The American Heritage Dictionary of the English Language.* Houghton Mifflin Company and American Heritage Incorporated, fourth edition edition.

[Karel Pala, Pavel Ševeček, 1997] Karel Pala, Pavel Ševeček (1997). Valence českých sloves. In *Sborník prací FFUB*, pages 41–54.

[Kingsbury et al., 2002] Kingsbury, P., Palmer, M., and Marcus, M. (2002). Adding Semantic Annotation to the Penn TreeBank. In *Proceedings of the HLT Conference '02*, San Diego.

[Kocek et al., 2000] Kocek, J., Kopřivová, M., and Kučera, K., editors (2000). *Czech National Corpus - introduction and user handbook (in Czech).* FF UK - ÚČNK, Prague.

[Král, 2004] Král, R. (2004). *Jaký to má význam?* PhD thesis, Masaryk University.

[Král, 2001] Král, R. (2001). Three approaches to word sense disambiguation for czech. In *TSD '01: Proceedings of the 4th International Conference on Text, Speech and Dialogue*, pages 174–179, Berlin. Springer-Verlag.

[Král, 2002] Král, R. (2002). Word sense discrimination for czech. In *TSD '02: Proceedings of the 5th International Conference on Text, Speech and Dialogue*, pages 155–158, Berlin. Springer-Verlag.

[Langley et al., 1992] Langley, P., Iba, W., and Thompson, K. (1992). An analysis of bayesian classifiers. In *National Conference on Artificial Intelligence*, pages 223–228.

[Lee and Ng, 2002] Lee, Y. K. and Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 41–48, Morristown, NJ, USA. Association for Computational Linguistics.

[Lopatková et al., 2005] Lopatková, M., Bojar, O., Semecký, J., Benešová, V., and Žabokrtský, Z. (2005). Valency Lexicon of Czech Verbs VALLEX: Recent Experiments with Frame Disambiguation. In *8th International Conference on TSD*, pages pp. 99–106.

[Lopatková et al., 2006] Lopatková, M., Žabokrtský, Z., and Skwarska, K. (2006). Valency lexicon of czech verbs: Alternation-based model. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1728–1733, Paris, France.

[Lopatková et al., 2003] Lopatková, M., Žabokrtský, Z., Skwarska, K., and Benešová, V. (2003). Vallex 1.0 valency lexicon of czech verbs. Technical report, ÚFAL MFF UK.

[Marcus et al., 1994] Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1994). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

[McCallum, 2002] McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

[McDonald et al., 2005] McDonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005). Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proceedings of HLT Conference and Conference on EMNLP*, pages pp. 523–530, Vancouver, Canada. ACL.

[Pala and Smrž, 2004] Pala, K. and Smrž, P. (2004). Building Czech Wordnet. *Romanian Journal of Information Science and Technology*, 7(1–2):pp. 79–88.

[Panevová, 1974] Panevová, J. (1974). On verbal frames in Functional generative description I. *Prague Bulletin of Mathematical Linguistics*, (22):3–40.

[Panevová, 1980] Panevová, J. (1980). *Formy a funkce ve stavbě české věty.* Prague:Academia.

[Panevová, 1994] Panevová, J. (1994). Valency frames and the meaning of the sentence. *The Prague School of Structural and Functional Linguistics*, pages 223–243.

[Petr Karlík, 2002] Petr Karlík, Marek Nekula, J. P., editor (2002). *Encyklopedický slovník češtiny.* Nakladatelství Lidové noviny. Fourth Edition.

[Pietra et al., 1997] Pietra, S. D., Pietra, V. D., and Lafferty, J. (1997). Inducing features of random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):380–393.

[Quinlan, 1986] Quinlan (1986). Induction of decision trees. *Machine Learning.*

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufman.

[Quinlan, 1996] Quinlan, J. R. (1996). Bagging, boosting, and c4.5. In *Proceedings, Fourteenth National Conference on Artificial Intelligence.*

[Quinlan, 2002] Quinlan, J. R. (2002). Data mining tools see5 and c5.0. http://www.rulequest.com/see5-info.html.

[Quiroga-Clare, 2003] Quiroga-Clare, C. (2003). Language ambiguity: A curse and a blessing. *Translation Journal.* http://accurapid.com/journal/23ambiguity.htm.

[Rivest, 1987] Rivest, R. L. (1987). Learning decision lists. *Machine Learning*, 2(3):229–246.

[Semecký, 2006] Semecký, J. (2006). On automatic assignment of verb valency frames in czech. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1941–1944, Paris, France.

[Semecký and Podveský, 2006] Semecký, J. and Podveský, P. (2006). Extensive study on automatic verb sense disambiguation in czech. In *Proceedings of the 9th International Conference, TSD 2006.*

[Sgall et al., 1986] Sgall, P., Hajičová, E., and Panevová, J. (1986). *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects.* Academia/Reidel Publishing Company, Prague, Czech Republic/Dordrecht, Netherlands.

[Vossen et al., 1998] Vossen, P., Bloksma, L., Rodriguez, H., Climent, S., Calzolari, N., Roventini, A., Bertagna, F., Alonge, A., and Peters, W. (1998). The eurowordnet base concepts and top ontology. Technical report, Centre National de la Recherche Scientifique, Paris, France, France.

[Štěpánek, 2006] Štěpánek, J. (2006). *Závislostní zachycení větné struktury v anotovaném syntaktickém korpusu (nástroje pro zajištění konzistence dat).* PhD thesis, FF UK - ÚČNK.

[Yarowsky, 1994] Yarowsky, D. (1994). Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In *Meeting of the Association for Computational Linguistics*, pages 88–95.

[Žabokrtský and Lopatková, 2004] Žabokrtský, Z. and Lopatková, M. (2004). Valency Frames of Czech Verbs in VALLEX 1.0. In Meyers, A., editor, *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 70—77, Boston. Association for Computational Linguistics.

[Žabokrstký, 2004] Žabokrstký, Z. (2004). *Valency Lexicon of Czech Verbs.* PhD thesis, Institute of Formal and Applied Linguistics, Charles University, Prague.