Czech language tagging

Barbora Hladká

Doctoral Thesis

Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University
Prague 2000

Opponents:

Prof. Frederick Jelinek, Johns Hopkins University, Baltimore
Prof. PhDr. Petr Sgall, DrSc., Charles University, Prague

Za lásku děkuji Tomášovi; za životní příklad děkuji svým rodičům; za přátelství děkuji Bibiáně a Veronice.

Za důvěru děkuji Janu Hajičovi; za to, že se po mém zaklepání vždy ozvalo 'dále', děkuji Evě Hajičové; za Summer Workshop'98 děkuji Fredericku Jelinkovi; za 'zapeklité' otázky, které padaly při pondělních seminářích, děkuji Petru Sgallovi.

Za horké maliny se zmrzlinou děkuji Aleně Böhmové a Kirilu Ribarovovi.

Za to, že tato práce vznikla, děkuji i všem ostatním kolegům a přátelům.

Abstract

Czech Language Tagging

Barbora Hladká
Supervisor: Jan Hajič

Corpus linguistics is interested in the way people use language in speech and writing; the usage is documented in a corpus. In general, a corpus is conceived of as a structured and annotated collection of texts covering written or spoken language resources. In terms of computational linguistics, a corpus is a huge, electronically (by computer) processed collection of texts and speeches containing a variety of information. Corpus discloses the factual usage of language patterns and represents a source for linguistic statistics. Using corpora for various linguistic tasks, we speak about corpus-based approaches.

Practically every natural language processing system (machine translation, information retrieval, parsing, etc.) for (not only) an inflective language needs a morphologically processed text, i.e. to know for each word the list of all possible combinations (tags) of morphological category values which make sense for the given word. However, most the systems need more precise information. They need just a single combination of morphological category values (from the list of all available combinations) to be identified which fits to the particular context. The task called tagging uses the context of a word in the input text to select the correct tag from the list of all possible tags.

The goal of this dissertation is to make progress toward tagging a highly inflective language, namely Czech. Since all currently used tagging approaches are driven by corpus-based methods, we can take advantage of having at our disposal a structured and annotated Czech corpus. This fact and the absence of corpora for other Slavic or similar languages promote the presented results as a pioneering effort with many positive and stimulating conclusions.

For a more sophisticated evaluation of tagging of Czech, we apply our code and settings also to tag English texts. Czech language exhibits a rich

inflection accompanied by a high degree of ambiguity. On the other hand, English represents a language with poor inflection. Differences between Czech and English (from the point of view of morphology) are reflected in the amount of information included in the tags. It is no wonder then that the performances of tagging systems on English are still better.

Our main focus is on the data and on the various corpus-based methods we apply to tag texts - hidden Markov model based approach, rule-based approach and exponential model based approach. We discuss in detail the results we have obtained when tagging Czech written texts.

A special attention is paid to the considerations on the context. Our aim is to concentrate on the idea which context should be selected from the processed text to tag it properly.

As the performances of Czech tagging systems are very close to each other in the end, we propose and test an original strategy of a combination of the tagging methods, to achieve better results.

To have more morphologically annotated data compatible with the Prague Dependency Treebank, we convert (by semi-automatic procedures) the original data available to pioneer Czech tagging experiments into the format of the Prague Dependency Treebank.

Contents

List of Tables

List of Figures

Chapter 1

Language Processing

1.1 Natural Language Processing

To specify what Natural Language Processing (NLP) means, we will trace, first, the senses of the lexicon [Collins Cobuild English Dictionary, 1995] head words corresponding to the words from the title (*natural, language, processing*) independently and we try then, to put together those senses which suit for our interpretation if we say "Natural Language Processing is . . . ".
The following relevant senses are assigned to the respective lexical head words:

- Someone with a **natural** ability or skill was born with that ability and did not have to learn it.
- A **language** is a system of communication which consists of a set of sounds and written symbols which are used by the people of a particular country or region for talking or writing. ... *the English language* ...
- when people **process** information, they put it through a system or into a computer in order to deal with it. ◊ **processing** ... *data processing* ...

It could seem meaningless to separate *natural* from *language*. Unfortunately, neither *natural* nor *language* do appear under the headwords *language* and *natural*, respectively in the lexicon from which we draw particular senses.

Putting together the chosen senses of *natural* and *language*, we get a "rough" meaning of the term *natural language*: it is a system of communication and people were born with ability and skills to talk and to write (i.e. to use this system) without any learning. But, in case of natural language, we are not sure whether one has to or does not have to acquize it. The question how children learn the language in reality is difficult to answer and remains still open.

The way the language is being used can be observed in different types of writing and speech. The books, newspapers, letters, magazines, conversations, interviews, meetings bring a lot of natural language information. Using these language resources, we can follow the usage of various language "events". How to follow them? To read an unimaginable number of written texts or to listen to (again an unimaginable number of) records. As we speak about an *era of computers*, let us join human forces with a computer.

Finally, we can answer the question *What is Natural Language Processing?* **Natural Language Processing** is an analysis of natural language information using a computer[1].

In principle, there are two basic ways how a computer can analyse language information:

(a) to encode human "know-how" directly into a computer program

(b) to simulate getting of human "know-how" and human applying of "know-how" by a computer program

To be more specific, we illustrate these two approaches to the problem of classification of words according to the part of speech criterion, i.e. for identification of nouns, verbs, pronouns, etc. in a sentence, in a chapter, in a book. This task is motivated by such questions as *Which part of speech is observed most frequently?* or *Are nouns preceded by adjectives more frequently than by pronouns?*

Given the approach (a) one pre-annotates, for instance, 20 sentences and gets (with certain simplifications) such observations as: if the word is "vedoucí" and is preceded by an adjective then it is a noun ("boss"); otherwise it is an adjective ("leading"). Consequently, such observations are encoded into a computer code which annotates whatever texts we need. It is an obvious fact that we need texts to be annotated as precisely as possible. The "quality" of annotation depends on the "quality" of observations. As the number of pre-annotated text increases, one gets a large number of observations which describe not only general events but specific events as well. On the other hand, we are sure that in principle the observations cannot be absolutely correct. Altogether, one has to pre-annotate a reasonable number of texts and to generate the observations. The computer then annotates texts using the observations. We speak about a **knowledge-based** approach.

The approach (b) provides a way how to minimize the amounts of human work and how to get observations automatically. One pre-annotates the

[1] In practice, the term *NLP* is used mostly to refer to written language resources and *speech recognition* to spoken language. Within the language processing introduction, we will concentrate on the NLP issues and terms related to the presented work.

texts and then the computer (not a human!) learns the observations from the pre-annotated texts. We call such an approach a **corpus-based approach**, where the pre-annotated text is understood as the corpus. The corpus-based approach exhibits a very useful tool to discover phenomena which no one could observe without examining unimaginable amount of data.

Corpus is a vast, electronically (by computer) processed, uniformly structured and continually added collection of language texts or speeches containing (not essentially) a variety of (as much explicit as possible) information the corpus might (implicitly) provide.

The corpus (or part of corpus only) ready for any linguistic application is usually split into two parts - the **training data** and the **test data**. In addition, it is often useful to separate so called **held-out data** as well. Ordering the parts according to their size, the training data should be the biggest one. The held-out data and the test data should be split relatively to the total size of the used corpus. The training and held-out data supply the data for training (learning) the events determined by the needs of the particular application. While the training data provide sources for training the basic algorithm, the held-out data offer the sources for tuning the parameters specifying the algorithm. The quality of the chosen algorithm is measured on the test data "deprived" of the annotations. Consequently, the output of algorithm is compared with the test data annotations.

The most wide-spread corpus-based methods are the **statistical** (or probabilistic) **methods**. The statistical methods offer good theoretical background, an automatic estimation of probabilities from data and a direct way how to disambiguate the particular information. To illustrate the statistical approach, let us have look at the problem of **parsing** (= a syntactical analysis of a sentence). Given any of the two core methodological approaches to the syntactic analysis - the approach based on the dependency structures and the approach based on the phrase structures - the parsing procedure returns a syntactic tree for the input sentence. To parse the test data using the statistical methods includes the following steps: (i) the syntactic annotation of the texts, (ii) the separation of the training and test data (no held-out data in the easiest case), (iii) the training of the probabilistic model (e.g. driven by the relative frequency of events) on the training data and finally (iv) the assignments (based on the trained probabilistic model) of the syntactic trees to the sentences in the test data.

1.2 Czech Language Processing

As Czech language is a natural language, we can simply reformulate our NLP definition so that **Czech Language Processing** (CzLP) is an analysis of Czech language information using a computer. Looking at the CzLP from the perspective of the usage of the annotated corpora, we can traditionally classify the CzLP approaches without the corpora as knowledge-based and the CzLP approaches via the corpora as corpus-based.

A method of automatic extraction of significant terms from texts (MOSAIC, [Kirschner, 1983]) provides for the input scientific texts the sets of expressions (simple, complex terms) which reflect the general theme of the input texts. The selection of the expressions to be extracted is based on linguistic criteria driven by the semantic properties in the morphemic structures. Experimental versions of MOSAIC were implemented for Czech and Slovak.

A complex description of a method designed for the build-up of a question-answering system (KODAS) capable to accept questions formulated in Czech and to retrieve answer in a simple database is given in [Hajič, 1984]. The KODAS was elaborated in close connection with the systems MOSAIC and TIBAQ.

ASIMUT - a system to automatically retrieve terms or whole terminological collocations in full texts (details in [Králíková, Panevová, 1990] - was originally designed for Czech. The ASIMUT is not based on any directed dictionary; the linguistic information necessary for lemmatization is added automatically and the potential user has to know the rules of the query language.

The Czech text-and-inference based approach to question answering is an automatic system, called TIBAQ, concentrating on human-machine communication. The idea is to construct a system as general as possible which can be easily modified for a specific domain on the basis of a serious linguistic analysis. The experiments (domain of electronics and endoscopic examination) conceive of a question answering system with questions and answers in Czech. The knowledge bases of the system reflect semantic-pragmatic representations of the input sentence and inference rules ("if" "then" form) processing of the given representations. A detailed description of presented approach is available in [Hajičová et al., 1995].

Naturally, the CzLP knowledge-based approaches are "older" than the corpus-based ones. At the same time, the use of the corpora does not mean an absolute exclusion of the knowledge-based approaches.

Grammar checkers are being developed in order to provide information about the type and location of grammatical errors within a sentence. The grammar checker represents one example of a practically oriented application. In [Kuboň, Holan, Plátek, 1997], the authors are discussing not only the theoretical background but also the implementation ideas of the grammar checker of Czech sentences.

The CzLP corpus-based research started to appear at the beginning of the 90s, firstly thanks to the existence of the Czech Tagged Corpus (Sect. 3.1.2) and later thanks to the building of the Prague Dependency Treebank (Sect. 3.1.5).

The rule based error-driven learning algorithm (described in [Brill, 1998]) was applied (besides other) for parsing free text (see [Brill, 1993b]). The modification of this corpus-based method in order to work with a dependency tree structure which describes more efficiently the syntax of the free-word order languages, such as the Czech language, is the major concern of the work explained in [Ribarov, 1996] and [Hajič, Ribarov, 1997]. We have already mentioned above the principle of a statistical approach to parse free text. A simple probabilistic model built over the dependency structure of Czech sentences is presented in [Zeman, 1998]. Both of the above mentioned Czech parsers (the rule-based and the statistical one) were motivated by the original efforts to parse English sentences and were designed for Czech language. However, there exists another idea how to parse Czech sentences - to take the parser originally designed for English and to modify it with regard to the character of the Czech language. The description of the practical experiments over this idea together with its results is described in [Hajič et al., 1998].

Besides the monolingual corpora, there emerge urgent demands for bilingual corpora, so much needed in machine translation applications. The pioneer statistical experiments of an automatic extraction of a Czech-English translation dictionary based on the bilingual Czech-English corpus are introduced in [Cuřín, Čmejrek, 1999].

Most of the enumerated CzLP issues call for preprocessing of their input text files in the sense of the morphological analysis. But for the most part, they need more precise information, a disambiguated morphological analysis. We provide, throughout this dissertation, a detailed description of steps we have carried out on the way from the first experiment on the automatic morphological analysis disambiguation ([Hladká, 1994]) up to the latest experiments.

Some aspects of computational solutions to formal Czech morphology was described in [Hajič, 1994]. A more comprehensive description of computational Czech morphology in connection with disambiguation of morphological analysis is given in [Hajič, in press].

Chapter 2

Tagging

In this chapter, we first present the tagging motivation (Sect. 2.1), illustrate possible tagging difficulties, we review then the key tagging corpus-based strategies and compare their basic characteristics (i.e. context handling, time and space requirements, performances, etc.), and, at the end, we give an overview of these strategies to tag English texts (Sect. 2.3).

2.1 Tagging Motivation

To tag something (or to mark something) by a specific kind of information embodied in *a tag*, we have to decide what exactly we mean by "something" and what we mean by a "specific kind of information". We have already introduced the main object of our interest - written natural language resources.

Tokenization is a process of splitting an input text document into units called **word form tokens**. Let a **word form** be a string of characters (letters, numbers) preceded and followed by a space or a delimiter such as a punctuation symbol. Abbreviations, acronyms and all kinds of numbers and punctuation marks are considered to be special types of word forms. More exactly, we work with **word form tokens**. Finally, let a word form token be an elementary unit we tag and for our purposes, we suppose that each input text to be processed is tokenized.

The next step which should be taken is to specify a repertoire of tags - a set of tags called *TAGS*. From the automatic processing point of view, in the course of tagging the tags should be designed in a unique and "economical" way: unique for expressing given information by exactly one tag, "economical" for a comfortable and "cheap" processing with regard to the design of data structure representing a tag. To tag word tokens in a sentence $w_{i=1}^{n}$, we take one word token after another and choose the appropriate tag from $TAGS$ for each word token according to the context preceding or following w_i. However, not all tags from $TAGS$ are meaningful for a word form w_i; it seems useful to select for each w the set $TAGS_w$ in such a way that it contains only the plausible tags for the word form w and is therefore a subset of *TAGS*. Then, the selection of the tags for word token w_i is limited to a particular set $TAGS_{w_i}$.

Formally, **the tagging procedure** ϕ selects a sequence of tags T for the input text W:

$$\phi : W \rightarrow T, \phi(w_i) = t_i, t_i \in TAGS_{w_i}, \forall i : 1 \leq i \leq n, n = |W|$$

where TAGS_{w_i} is the set of meaningful tags for a word token w_i.

So far, we have used the term tag in a general way. In the current usage in computational linguistics, the term tagging is used mainly for the assignment of part of speech information to a word token. Whereas this might be sufficient for morphologically impoverished languages such as English, we need and want more: natural language processing of a highly inflectional language such as Czech requires that tags contain the information given by natural language morphology, i.e. a tag embodies the values of morphological categories (MCs). The granularity of the set TAGS depends mainly on the task for which the tagging is used; in principle, we can freely modify TAGS from a very coarse granularity to a very detailed one or vice versa. The set $TAGS_w$ for a word w is determined by morphological analysis (MA).

In the sequel, when speaking about **tag** or **tagging**, we have strictly in mind the morphological point of view; even more, by tagging we mean an automatic assignment of tags covering morphological information. At the same time, the manual assignment of tags covering morphological information is called here an **annotation**[1]. We will indicate other cases of tagging or annotation explicitly. We can split the view on tags into two levels: on the **tag level**, we take tags as they are (i.e., as an "atomic" value), and on the **subtag level**, we "dig into" the tag and take each MC separately.

Tagging Difficulties

To illustrate what kind of difficulties we can face during the tagging, let us take a really "spicy" Czech sentence to be tagged:

Kose	trávník,	viděl	kose,	podíval	se	na	ně	kose.
Mowing	lawn	he-saw	baby-blackbird	he-looked	Refl.	at	it	wryly.

'Mowing lawn, he saw a baby blackbird, and looked at it wryly.'

Let us define three very simple tag sets (TAGS(A), TAGS(B), TAGS(C)) covering the following morphological information:

[1] In the sequel, we will speak about the Czech Tagged Corpus because of the historical convention though it is an annotated corpus.

(A) `part of speech` only

(B) `part of speech` and `case`

(C) `part of speech` and `gender`

Having in mind the economical and unique format of tags, we use the character (letter and numeral) representation of morphological category values. Tab. 2.1 contains all possible values of MCs we are interested in - 10 basic `part of speech` classes + 1 punctuation class; 7 + 1 `case` values and 4 + 1 `gender` values. Together, the tag set TAGS(A) contains 11 tags; the tag set TAGS(B) contains up to 77 (number of all possible combinations of `part of speech` and `case` values (excluding '-' value) = 11∗7) tags and tag set TAGS(C) up to 44 (11∗4) tags. However, not all combinations are meaningful. For instance, it does not make sense to determine `case` of verb, of conjunction, of adverb, or `gender` of preposition, of particle and so on. In case of meaningless combinations, we set up one more additional value '-' for those MCs which are not involved in the inflection of the particular part of speech class.

`POS`		`case`		`gender`	
value	description	value	description	value	description
A	adjective	1	nominative	M	masculine animate
N	noun	2	genitive	I	masculine inanimate
P	pronoun	3	dative	F	feminine
C	numeral	4	accusative	N	neuter
R	preposition	5	vocative	-	not specified
V	verb	6	locative		
J	conjunction	7	instrumental		
T	particle	-	not specified		
K	interjection				
D	adverb				
Z	punctuation				

Table 2.1: Motive selection of morphological categories

Reading along the lines in Tab. 2.2, the tag subsets (in the columns marked by 'TAGS(X)' (X $\in$ {A, B, C}) that are the output of MA in the frame of a given tag set are listed for the given word (totally, there are 9 word forms (12 word tokens) in the input sentence)[2].

[2]Thus e.g. the word form "na" could be tagged as a preposition ($\mathrm{TAGS}_{na}(\mathrm{A}) = \{\mathrm{R}\} \subseteq \mathrm{TAGS}(\mathrm{A})$), as a preposition connected with a noun in accusative or locative ($\mathrm{TAGS}_{na}(\mathrm{B})$

WORD FORM	TAGS(A)	TAGS(B)	TAGS(C)
kose	N,V,D	N1,N3,N4,N5,N6, V-,D-	NM,NN,NF, VF,VM,VI,VN,D-
trávník	N	N1,N4	NI
viděl	V	V-	VM,VI
podíval	V	V-	VM,VI
se	R,T	R7,T-	R-,T-
na	R	R4,R6	R-
ně	P	P4	PN,PF,PM,PI
,	Z	Z-	Z-
.	Z	Z-	Z-

Table 2.2: Motive tag sets

With respect to the size of the individual subsets, the word form"kose" seems to be the most complicated one. However, it is a common fact when a word token appears in the context of the other words, the ambiguity is often reduced. The ambiguity is often reduced, not completely removed! As expected, from the tagging point of view we 'like' the word forms for which there exists just a single tag, i.e. these word forms need no context to find their proper tag. Since with word forms which represent an ambiguity problem (the tag subset assigned to them contains more than one element), the context must take part in the process of tagging. However, in general, there is no strict rule saying how many preceding and following word tokens we should look at to be sure that we tag the word token properly. The criterion of the context dimension depends in fact on the character of the word order of the given language. More detailed considerations on the context are the main topic of the Chapter 4.

Observing the column marked by TAGS(B) in Tab. 2.2, we see that the word forms "kose", "trávník", "se" and "na" belong to the ambiguous word forms. The given sentence is being tagged from the left to the right and the ambiguous word tokens become unambiguous using the annotated corpus which represents a source of language usage. For instance, the word token"se" is preceded by an unambiguous verb; taking into account (as context information) the preceding word token together with its tag (podíval,

---

$= \{R4, R6\} \subseteq$ TAGS(B)) or a preposition for which `gender` is not involved in its inflection (TAGS$_{na}$(C) $= \{R\text{-}\} \subseteq$ TAGS(C))

V-), there is a strong preference to tag "se" as a reflexive particle (T-). On the other hand, taking into account just the tag of the preceding word token(the only information we have is the fact that "se" is preceded by verb), it is not so evident how to tag "se"; the decision depends on the frequency of the situation *verb followed by a preposition* and the situation *verb followed by a reflexive particle* in the annotated corpus.

As we are the witnesses of "blackbird's" event we give the appropriate MC values (in the order `part of speech, case, gender`) of word tokens in the input sentence: Kose/V-M trávník/N4I ,/Z-- viděl/V-M kose/N4I ,/Z-- podíval/V-M se/T-- na/R4- ně/P4I kose/D--.

2.2 Tagging Measures

To measure the quality of a tagger A, we use the usual **tagging accuracy (TA)** measure which gives *the percentage of correctly tagged words*:

$$TA(A) = (Correctly\_Tagged\_Words\_by\_A/Total\_Tagged\_Words) * 100(\%) \tag{2.1}$$

We can also express the performance of a tagger A by the **error rate (ER)** measure which gives *the percentage of incorrectly tagged words*:

$$ER(A) = (Errors\_Produced\_by\_A/Total\_Tagged\_Words) * 100(\%)$$
$$(\equiv ER(A) = 100 - TA(A)(\%)) \tag{2.2}$$

To illustrate the defined measures, let us assume a sentence $W = w_1$ w_2 w_3 w_4 w_5 and let us represent the output of the tagger as a sequence of *0*'s and *1*'s where *0* corresponds to an incorrectly tagged word and *1* to a correctly tagged word. The tagger A correctly tags words w_2 and w_4, the tagger B correctly tags words w_3, w_4 and w_5. Thus, we can express the output of the tagger A as a sequence T_A = *01010* and the output of the tagger B as a sequence T_B = *00111* and we can say that *Total_Tagged_Words* = 5, *Errors_Produced_by_A* = 3 (number of 0s in T_A, w_1, w_3, w_5), *Errors_Produced_by_B* = 2 (number of 0s in T_B, w_1, w_2). Finally, TA(A) = (2/5)∗100(%) = 40%, TA(B) = (3/5)∗100(%) = 60%. The conclusion is that the performance[3] of the tagger B is better than the performance of the tagger A.

[3] Speaking about the performance of a tagger we always have in mind the tagging accuracy of the tagger.

2.3 Corpus-Based Methods of Tagging

Several approaches to the automatic tagging of texts have been proposed. The so called *stochastic strategies* use various statistical models, namely Markov models (MM), the maximum entropy (ME) model and the exponential (EXP) model. A *memory-based* (MB) strategy represents a kind of supervised learning based on similarity-based reasoning. In a *rule-based* (RB) strategy, a set of meaningful rules is automatically acquired. *Neural networks* (NE) represent an artificial intelligence strategy. The strategies mentioned up to now belong to *corpus-based* approaches, i.e. they work on annotated corpora to achieve appropriate probabilities, memory patterns, transformation rules and weights. Table 2.3 provides the review of the results of representative corpus-based methods applied for English language.

Markov Model Tagger Let us formulate the problem of tagging in terms of (in)dependent random variables, stochastic (random) processes, Markov models, etc. We suppose that the set of possible tags TAGS for the given language is already designed.

A **stochastic** or **random process** is a sequence of random variables based on the same sample space Ω. The possible outcomes of variables constitute possible states of a stochastic process. In general, the random variables are independent. However, the variables are dependent in the course of the stochastic process.

Let us work with random variables τ_1, τ_2,..., for which Ω = TAGS = $\{t_1, t_2, ..., t_\tau\}$ is a discrete finite set of τ_j outcomes ($j \geq 1$), and likewise with random variables ω_1, ω_2,... for which Ω = LEXICON = $\{w_1, w_2, ..., w_\omega\}$ is a discrete finite set of ω_k outcomes ($k \geq 1$).

Let us consider an annotated paragraph W that contains n words so that ω_1 is the first word of the paragraph, ω_2 is the second word,... Let τ_1 be the tag of the first word, τ_2 the tag of the second word,..., τ_n the tag of the last word of the given paragraph. The sequence of the tags T creates a random process (states = tags) with discrete random variables and a discrete time parameter (we suppose that each word is "processed" in every time unit).

A **Markov model** (MM) is a stochastic process where the probability of the next state given the entire sequence of previous states up to the current state is dependent only on the current state (this is called **Markov property** or the **first Markov assumption**) (see Eq. 2.3)

$$p(t_{i_{t+1}}|t_{i_1}, \ldots, t_{i_t}) \approx p(t_{i_{t+1}}|t_{i_t}) \qquad (2.3)$$

To apply the Markov property on tagging means that the probability that the (t+1)-th word is marked by the tag $t_{i_{t+1}}$ depends only on the tag of the previous word and not on tags of all previous words "read" up to the current point. The probability of a Markov model t_{i_1}, t_{i_2},...,t_{i_n} can be expressed as (3.4) (using the Bayes' formula $p(A|B) = p(A,B)/p(B)$)

$$p(t_{i_1}, t_{i_2}, ..., t_{i_n}) \approx p(t_{i_1}) * p(t_{i_2}|t_{i_1}) * p(t_{i_3}|t_{i_2}) * \ldots * p(t_{i_n}|t_{i_{n-1}}) \tag{2.4}$$

The conditional probabilities $p(t_{i_{t+1}}|t_{i_t})$ are called the transition probabilities of the Markov model, i.e. the probability of the transition from the state t_{i_t} to the state $t_{i_{t+1}}$. In the tagging terminology, the transition probabilities are called the **contextual probabilities** and the current context for the tag $t_{i_{t+1}}$ of the *(t+1)*-th word is determined by the tag t_{i_t} of the t-th word.

The Markov models for which the probability of the next state depends on the current state are called Markov models of the **first order**. In the Markov models of an m**-th order** the probability of the next state depends on the m previous states and the states correspond to a sequence of *m-1* tags. **The second Markov assumption** says that the probability of a particular output depends only on the current state and not on the sequence of the previous states and the previous outputs:

$$p(w_t, t_{i_t}|w_1, w_2, ..., w_{t-1}, t_{i_1}, t_{i_2}, ..., t_{i_{t-1}}) \approx p(w_t|t_{i_t}) \tag{2.5}$$

In the tagging terminology, the conditional probabilities $p(w_t|t_{i_t})$ are called **lexical probabilities**.

If it is not possible to observe the sequence of states of a Markov model, but only the sequence w_1, w_2,..., w_n of output signals, (n is the size of the paragraph to be tagged), the model is called a **hidden Markov model** (HMM).

Let W be a sequence of output signals (which we know) and let T be an unknown sequence of states in which W is produced. The goal is to find such a sequence of states Γ that maximizes the probability of the sequence T given the W. The formula expressing the condition put on the optimal state sequence is

$$\Gamma = \max_T p(T|W) \tag{2.6}$$

Given the Bayes' formula, we can rewrite the formula (2.6) as

$$\Gamma = \max_T (p(W|T) * p(T))/p(W) \tag{2.7}$$

As $p(W)$ does not depend on T and it is a positive number, we can approximate

$$\Gamma \approx \max_T p(W|T) * p(T) \tag{2.8}$$

Using the Markov assumptions (Eqs. 2.3 and 2.5) the equation expressing the condition put on the optimal state sequence of the first order Markov model (**bigram Markov model**) is

$$\Gamma \approx \max_T p(w_1|t_{i_1}) * \prod_{t=2}^{n} p(w_t|t_{i_t}) * p(t_{i_t}|t_{i_{t-1}}), T = t_{i_1}, t_{i_2}, ..., t_{i_n} \tag{2.9}$$

Similarly for the second order Markov model (**trigram Markov model**[4]):

$$\Gamma \approx \max_T p(w_1|t_{i_1}) * p(t_{i_1}) * p(t_{i_2}|t_{i_1}) * \prod_{t=3}^{n} p(w_t|t_{i_t}) * p(t_{i_t}|t_{i_{t-1}}, t_{i_{t-2}}),$$
$$T = t_{i_1}, t_{i_2}, ..., t_{i_n} \tag{2.10}$$

In practice, we work also with the so called **unigram Markov model**, which does not take into account any history and is just looking for the most probable tag for given word:

$$\Gamma \approx \max_T \prod_{t=1}^{n} p(w_t|t_{i_t}), T = t_{i_1}, t_{i_2}, ..., t_{i_n} \tag{2.11}$$

If an annotated corpus is available (W_{train}, T_{train}), i.e. if we know the sequence of the output signals (= words, W_{train}) and the sequence of the states (= tags, T_{train}), we can calculate (train) the distribution $p(T_{train})$ and $p(W_{train}|T_{train})$ directly from the observed relative frequencies in the annotated corpus, i.e. we can directly construct the corresponding Markov model.

To tag a non-annotated text W_{test} means to determine the most probable tag sequence Γ_{test} that suits Eqs. 2.9 and 2.10 calculating with the trained MM determined by the distribution $p(T_{train})$ and $p(W_{train}|T_{train})$. In order

[4]**Tag N-gram** in the annotated corpus is a sequence of N tags which follow one another. For $N = 2$, we speak about a **tag bigram** and for $N = 3$ about a **tag trigram**.

Let us illustrate these notions on the annotated sentence from Section 2.1: Kose/V-M trávník/N4I ,/Z-- viděl/V-M kose/N4I ,/Z-- podíval/V-M se/T-- na/R4- ně/P4I kose/D--. The list which follows contains all different tag bigrams and tag trigrams together with their counts within the annotated sentence. For instance, tag bigram (V-M, N4I) occurs twice; (V-M, N4I; 2), (N4I, Z--; 2), (Z--, V-M; 2), (V-M, T--; 1), (T--, R4-; 1), (R4-, P4I; 1), (P4I, D--; 1); tag trigrams (V-M , N4I , Z--; 2), (N4I , Z-- , V-M ; 1), (Z-- , V-M , N4I ; 1), (N4I , Z-- , V-M ; 1), (Z-- , V-M , T--; 1), (V-M , T-- , R4-; 1), (T-- , R4- , P4I ; 1), (R4- , P4I , D--; 1).

to prevent the problem of unknown words which occur in W_{test} and do not occur in W_{train}, i.e. $p(unknown\_word|tag) = 0, \forall tag : tag \in TAGS$, we use a linear smoothing to generate the distribution $\widetilde{p}(T_{train})$ and $\widetilde{p}(W_{train}|T_{train})$ instead of the trained one so that $\widetilde{p}(word|tag) > 0$ for any word and any tag from the set *TAGS*. The Viterbi algorithm is used to find an optimal sequence of tags and employs the parameters $\widetilde{p}(T_{train})$, $\widetilde{p}(W_{train}|T_{train})$ and the sets $TAGS_w$ (see Sect. 2.1). The complexity of the Viterbi searching of the optimal tag sequence is $\mathrm{O}(N_{test} * T^2)$, where N_{test} is the test data size and T is the number of states.

It is not our objective to explain the smoothing procedure and the Viterbi algorithm in detail; for a detailed explanation see [Jelinek, Mercer, 1980] and [Forney, 1973]. We only present here the final equations (including the smoothing and relative frequency counting) expressing the condition imposed on the optimal tag sequence Γ for all mentioned MMs we discuss, i.e. unigram MM, bigram MM and trigram MM:

$$\Gamma \approx \max_T \prod_{t=1}^{n} \widetilde{p}(w_t|t_{i_t}), T = t_{i_1}, t_{i_2}, ..., t_{i_n} \tag{2.12}$$

$$\Gamma \approx \max_T \widetilde{p}(w_1|t_{i_1}) * \prod_{t=2}^{n} \widetilde{p}(w_t|t_{i_t}) * \widetilde{p}(t_{i_t}|t_{i_{t-1}}), T = t_{i_1}, t_{i_2}, ..., t_{i_n} \tag{2.13}$$

$$\begin{aligned}\Gamma \approx \max_T \widetilde{p}(w_1|t_{i_1}) * \widetilde{p}(t_{i_1}) * \widetilde{p}(t_{i_2}|t_{i_1}) * \prod_{t=3}^{n} \widetilde{p}(w_t|t_{i_t}) * \widetilde{p}(t_{i_t}|t_{i_{t-1}}, t_{i_{t-2}}), \\ T = t_{i_1}, t_{i_2}, ..., t_{i_n}\end{aligned} \tag{2.14}$$

where

$$\widetilde{p}(w_t|t_{i_t}) = \lambda_w * p(w_t|t_{i_t}) + (1 - \lambda_w) * 1/W_{t_{i_t}} \tag{2.15}$$

$$\widetilde{p}(t_{i_t}) = \lambda_{01} * p(t_{i_t}) + (1 - \lambda_{01}) * 1/C_T \tag{2.16}$$

$$\widetilde{p}(t_{i_t}|t_{i_{t-1}}) = \lambda_{11} * p(t_{i_t}|t_{i_{t-1}}) + \lambda_{12} * p(t_{i_t}) + (1 - \lambda_{11} - \lambda_{12}) * 1/C_T \tag{2.17}$$

$$\begin{aligned}\widetilde{p}(t_{i_t}|t_{i_{t-1}}, t_{i_{t-2}}) = \lambda_{21} * p(t_{i_t}|t_{i_{t-1}}, t_{i_{t-2}}) + \lambda_{22} * p(t_{i_t}|t_{i_{t-1}}) + \lambda_{23} * p(t_{i_t}) + \\ +(1 - \lambda_{21} - \lambda_{22} - \lambda_{23}) * 1/C_T\end{aligned} \tag{2.18}$$

$$p(w_t|t_{i_t}) = Count(w_t, t_{i_t})/Count(t_{i_t}) \tag{2.19}$$

$$p(t_{i_t}) = Count(t_{i_t})/|T_{train}| \tag{2.20}$$

$$p(t_{i_t}|t_{i_{t-1}}) = Count(t_{i_t}, t_{i_{t-1}})/Count(t_{i_{t-1}}) \tag{2.21}$$

$$p(t_{i_t}|t_{i_{t-1}}, t_{i_{t-2}}) = Count(t_{i_t}, t_{i_{t-1}}, t_{i_{t-2}})/Count(t_{i_{t-1}}, t_{i_{t-2}}) \tag{2.22}$$

where $W_{t_{i_t}}$ is the number of words that have the tag t_{i_t}, C_T is the number of different tags in T_{train}, $\lambda_w, \lambda_{01}, \lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22}, \lambda_{23} \leq 1$ and *Count(x)* is the frequency of an event x in the training text.

In [Merialdo, 1994], the author provides a very convincing comparison of the taggers based on Markov models (i.e. with an annotated corpus) and on hidden Markov models (without an annotated corpus). His experiments confirm the assumption that the more annotated texts are available the better training is obtained. The tagger trained on 955k words of the annotated Associated Press corpus (76 tags) has the tagging accuracy 97%.

Maximum Entropy Tagger described in [Ratnaparkhi, 1996] "manipulates" with a probabilistic model basically defined as $p(t, x) = \pi\mu \prod_{i=1}^{n} \lambda_i^{f_i(t,x)}$, where x is a context from the set of possible word and tag contexts, t is a tag from the set of possible tags, π is a normalization constant, $\{\mu, \lambda_1, \lambda_2, ..., \lambda_n\}$ are the positive model parameters and $\{f_1, f_2, ..., f_n\}$ is a set of yes/no features; i.e. f_i(t,x) $\in$ {0, 1}. Each parameter λ_i (the so called *feature weight*) corresponds to exactly one feature f_i and features operate over the events (context, tag). For a currently processed word, the set of specific contexts is limited to the currently processed word, the preceding two words together with their tags and the following two words. The positive model parameters are chosen (according to the MLP) to maximize the likelihood of the training data.

During the test step, the tagging procedure gives for each word a list of B highest probability sequences up to and including the currently processed word. The performance of the baseline model for English is 96.6% (training data size is 962kB) and the experiment uses the size B = 5. The complexity of the searching procedure is O($N_{test} * T * F * B$), where N_{test} is the test data size (number of words), T is the number of meaningful tags, F is the average number of features that are active for the given event (t,x) and B is explained above. The cost of the parameter estimation is O($N_{train} * T * F$), where T, F are defined above and N_{train} is the training data size.

Exponential Tagger was first introduced in [Hajič, Hladká, 1998b] and is described in detail in [Hajič, in press]. EXP approach is primarily designed for Czech tagging. It predicts proper tags from the list of meaningful tags given by AMA which works with a positional tag system (see Par. 3.1.5). With respect to the tag and subtag levels, the ME tagger (described above) operates on the tag level, whereas the EXP tagger operates on the subtag level. The ambiguity on the subtag level is mapped onto the so called *ambiguity classes* (ACs). For instance, for the word "se" the morphology generates two possible tags (i.e. the case of ambiguity on the tag level) RV--7---------- (preposition "with") and P7-X4---------- (reflexive particle). The ambiguity on the subtag level is represented by four ACs: [RP] (1st subtag), [V7] (2nd subtag) [-X](4th subtag) and [74] (5th subtag). The number of ACs matches the number of MCs, the value of which is not unique across the list of tags for a given word.

With regard to the ACs, the EXP tagger generates a separate model $p_{AC}(y|x)$ (where x is a context, y is the predicted subtag value $\in$ Y), which has the general form ([Berger et al., 1996]) determined by the Eq. 2.23 for each AC, while the ME tagger "manipulates" with just one probability distribution p.

$$p_{AC,e}(y|x) = \frac{\exp(\sum_{i=1}^{n} \lambda_i f_i(y,x))}{Z(x)} \tag{2.23}$$

where Z(x) is the normalization factor:

$$Z(x) = \sum_{y \in Y} \exp(\sum_{i=1}^{n} \lambda_i f_i(y,x)) \tag{2.24}$$

To avoid the "null probabilities" $p_{AC}(y|x)$ caused by an unseen context in the training data or by an unseen AC in the training data (i.e. there is no model for it), we formulate the final $p_{AC}(y|x)$ distribution using the smoothing procedure:

$$p_{AC}(y|x) = \sigma p_{AC,e}(y|x) + (1-\sigma)p(y) \tag{2.25}$$

where p(y) is the unigram distribution per MC. $\{f_1, f_2, ..., f_n\}$ is a set of yes/no features; i.e. $f_i(y,x) \in \{0, 1\}$. Each parameter λ_i (so called *feature weight*) corresponds to exactly one feature f_i and the features operate over the events (subtag value, context). To define feature function more exactly, we have to introduce the following terms: Let Cat_{AC} be the ambiguity class AC of a morphological category Cat (for instance, $Cat = gender$ and $Cat_{AC} = \{feminine, neuter\}$), y be an attribute for the subtag value being predicted, x be an attribute for the context value and $\bar{y}$, $\bar{x}$ be values of y, x attributes. Thus,

the feature function $f_{Cat_{AC},\bar{y},\bar{x}}(y,x) \to \{0,1\}$ is well-defined iff

$$\bar{y} \in Cat_{AC} \tag{2.26}$$

The value of a well-defined feature function $f_{Cat_{AC},\bar{y},\bar{x}}(y,x)$ is determined by

$$f_{Cat_{AC},\bar{y},\bar{x}}(y,x) = 1 \Leftrightarrow y = \bar{y} \wedge \bar{x} \subseteq x \tag{2.27}$$

The computing of the feature weights is usually based on the maximum entropy approach originally described in [Berger et al., 1996]. However, in the exponential approach, the weight estimation is built on the ratio of conditional probability of y in the context defined by the feature f$_{AC,\bar{y},\bar{x}}$ and the uniform distribution for the ambiguity class AC, $\lambda_{f_{AC,\bar{y},\bar{x}}} = \frac{p_{AC}(y|\bar{x})}{1/|AC|}$. While the ME tagger uses the MLP for selection of the features, the EXP tagger puts stress on the model's feature selection (during the training step) from the error rate point of view (like in the RB approach, see below). From the pool of features available for selection those features are chosen which lead to the maximal improvement in the error rate with respect to the setting of the threshold. The threshold is set to half the number of data items which contain the ambiguity class AC at the beginning of the feature selection loop, and then it is cut in half at every iteration.

The designed algorithm predicts all MCs independently and even more, the prediction is based on the ACs rather than on the previously predicted values. Thus, the tag which is given by the EXP tagger does not have to be an element of the list of tags returned by the AMA for the given word. That is why, the purely subtag independent strategy is modified by the so called Valid Tag Combination (VTC) strategy. The formula 2.28 expresses the dependence assumption.

$$p(t|x) = \prod_{Cat_{AC},Cat \in Categories} p_{AC}(y_{AC}|x) \tag{2.28}$$

where t is a complete tag, x is a context, $y_{AC} \in Cat_{AC}$ and p_{AC} is determined by Eq. 2.25.

Let N_{train} be the training data size, N_{test} be the test data size, F be the average number of active features for the given event (y,x), A be the average number of ACs for the given word and C be the average number of AC elements. Then the complexity of training is O$(N_{train} * F * A * C)$ and the complexity of tagging of the test data is O$(N_{test} * F_{final})$, where N_{test} is the test data size and F_{final} is the number of trained features.

Penn Treebank has been used for the EXP tagging of English texts. A Penn Treebank tag set has been converted to a positional tag system including the usage of subtag value letters defined in the frame of the Czech positional tag system. Penn Treebank positional tag is defined as a concatenation of 4 MCs - `part of speech, subpart of speech, number, gender`. For instance, for the word "*under*" the morphology generates three possible Penn Treebank tags: IN (preposition), JJ (adjective) and RB (adverb). The given positional tags are RR--, AA-1, DO-1. The EXP tagger trained on WSJ (1.2M words) gives 96.8% TA.

Memory-Based Tagger is introduced in [Daelemans, Zavrel, 1996]. As mentioned above, the MB tagger represents a supervised learning that makes use of similarity-based reasoning. At the same time, the MB tagger belongs to the corpus-based taggers, which work on an annotated corpus. Based on such a corpus, the MB tagger "maps" training examples into three structures: lexicon, case base for known words and case base for unknown words. Each training pattern in the case bases includes *word*, *manually assigned tag* and *context information*. During testing, for each test pattern (*word*, *context information*) its distance from all training patterns (belonging to the case base specified according to the presence of the test word in the lexicon) present in the memory is computed and a tag from "the closest" training pattern is assigned to the given word in the test data.

In particular, patterns (training, test) are represented as a vector of feature values for each particular tag. There are two important issues which have to be discussed: the distance metric and the representation of training patterns in the memory. The metric (Δ_{IB-IG}) ([Daelemans, Zavrel, 1996]) takes into account not only the comparison of appropriate feature values of two patterns but also the information gain of the feature value. These patterns are represented in the memory as *IGTrees* ([Daelemans, Zavrel, 1996]), which provide a comfortable way of an automatic identification of the optimal context size.

The complexity of a searching test pattern in a tree is O($\log(V) * F$), where F is the number of features (= the maximal depth of tree) and V is the average number of values per feature (= the average branching factor in the tree). The cost of the building of the tree is O($N_{train} * \log(V) * F$), where N_{train} is the number of training patterns (training data size), F and V are defined above. The *IGTree* implementation (*IGTrees,* Δ_{IB-IG}) of memory-based learning on English tagging has tagging accuracy 96.4% (trained on 2MB words, tested on 200kB).

Rule-Based Tagger The supervised transformation-based error-driven learning method described in [Brill, 1998] is classified as corpus-based; however, we have to stress that it employs not only a small annotated corpus but a large unannotated corpus as well. A pool of allowable lexical and contextual transformations is predetermined by templates operating on word forms and word tokens, respectively. A general lexical/contextual template has the form: *"for a given word change tag A to tag B if precondition C is true"* There are three main steps in the training process:

(a) From the annotated corpus, a lexicon is built specifying the most likely tag for the given word. The unknown words are tagged by the most frequently occurring tag in the annotated corpus.

(b) Lexical transformations are learned to guess the most likely tag for the unknown words (i.e. words not covered by the lexicon). The preconditions strictly oriented on the adding/deleting of prefixes/suffixes (resulting in the currently processed word form) and on the presence of a "special" character in the word form are crucial for these lexical transformations.

(c) Contextual transformations are learned to improve tagging accuracy. While lexical transformations operate over word forms, preconditions covering word tokens context are fundamental for contextual transformations.

It remains to explain why error-driven learning is employed. The learning procedure is carried out by iterations. During each iteration, the result of each transformation (an instantiation of a template) is compared to the truth and the transformation that causes the greatest error reduction is chosen. If there is no such transformation or if the error reduction is smaller than a specified threshold, the learning process is halted. The complexity of learning the cues is O(L$*N_{train}*$R), where L is the number of prespecified templates, N_{train} is the size of training data and R is the number of possible template instances. The complexity of tagging of the test data is O(T$*N_{test}$), where T is the number of transformations and N_{test} is the test data size. The rule-based tagger trained on 600K of English text has tagging accuracy 96.9%.

In [Megyesi, 1999], the author has demonstrated how Brill's rule-based tagger can be applied to a highly agglutinative language - Hungarian. When she applied a rule-based tagger as it is (i.e. designed for English), the tagging accuracy for Hungarian was not so high as for English (85.9% vs. 96.9%). To get a higher accuracy, the author modified lexical and contextual templates with regard to the character of Hungarian. After the modification of the templates, the tagging accuracy for Hungarian increased to 91.9% (training data size is 99,860 words; tag set size is 452).

Neural Network Tagger The tagger presented in [Schmid, 1994] is based on neural networks; it consists of a multilayer perceptron (MP) network and a lexicon. The given MP network contains only an input and an output layer and no hidden layers; further experiments indicated that a neural network tagger does not make a gain on a hidden layer. Each unit of the output layer of the MP network corresponds to one tag from the tag set. The context which is the input of the MP network includes the currently processed word, p number of previous words and f number of following words. MP network is trained on an annotated corpus using the so called backpropagation procedure. The lexicon has three parts - full form lexicon, suffix lexicon and a default entry; each of three parts covers a priori tag probabilities for each lexicon entry. The input-output layer version of tagger was trained on English text of 2 million words, the context was set up on 3 previous words, 2 following words and the number of training cycles was 4 millions. The performance of neural-network tagger is 96.2%.

Corpus-Based Taggers on English Our comparison of the representative corpus-based tagging strategies applied to English can be summarized by means of Tab. 2.3; it shows that the tagging accuracy of the individual approaches applied to English falls within a narrow range.

STRATEGY	TAGGER ID	TRAINING DATA (SIZE)	TAGGING ACCURACY (%)
Trigram MM ([Merialdo, 1994])	MM_EN	Associated Press (955Kw)	97.0
ME ([Ratnaparkhi, 1996])	ME_EN	WSJ (962Kw)	96.6
EXP ([Hajič, Hladká, 1998b])	EXP_EN	WSJ (1.2Mw)	96.8
MB ([Daelemans, Zavrel, 1996])	MB_EN	WSJ (2Mw)	96.4
RB ([Brill, 1998])	RB_EN	WSJ (600Kw)	96.9
NE ([Schmid, 1994])	NE_EN	WSJ (2Mw)	96.2

Table 2.3: Performances of the representative corpus-based tagging strategies applied to English

Chapter 3

Czech Tagging

In the previous chapter, we have discussed the corpus-based tagging strategies. The following issues belong to the most discussed questions: which methods to use to tag texts, which tag set is optimal, whether a bigger tag set or a smaller one is preferable, and how the tagging accuracy changes as the size of the tag set or the method of tagging changes.

In the present chapter, we will try to find out how these questions can be answered for Czech. We can take the advantage of having at our disposal rich (from the point of view of other Slavonic languages an unusually rich) set of annotated language resources. Each of the Czech annotated corpora is connected with a particular tag set. Concretely, we will describe four Czech tag sets - the Czech Tagged Corpus tag set, the Reduced Czech Tagged Corpus tag set, the positional tag set, and the Xerox tag set (Sect. 3.1).

Before performing any tagging experiments on Czech we first try to find out by means of the Czech and English annotated corpora if tagging of Czech is different from tagging of English (Sect. 3.2). For a more sophisticated answer, we apply the same code and settings to tag Czech and English texts (Sect. 3.4.1). We give a short introduction to the Czech automatic morohological analysis in Sect. 3.3.

Sect. 3.4 of this chapter is devoted to a detailed analysis of all results we obtained from various tagging experiments.

Describing all Czech tagging experiments we wished to identify them by unique titles which should express the basic information (tagging strategy, corpus, training data size, morphological preprocessing) regarding the experiments. As the basic information is too complex (we did not find any efficient way to code it into the experiment identification), we decided to use as the unique experiment IDs city names (written by the SansSerif font) to remind us of the factors relevant for the objective of this dissertation.

3.1 Language Resources

For the experiments described herein, we have used two different corpora: one “old” (Czech Corpus (CC) - texts from the 60s and early 70s, see

Sect. 3.1.1), one "new" (Czech National Corpus (CNC) - of a smaller volume but modern and technically compatible with our MA system, see Sect. 3.1.4). In particular, we have not used CC as it is, but we have converted it into the so called Czech Tagged Corpus (CTC), see Sect. 3.1.2. CNC represents the source of textual data for Prague Dependency Treebank (PDT, Sect. 3.1.5) and for Xerox Czech annotated data (Sect. 3.1.6). Together with the presentation of language resources we pay attention to the description of the designed tag sets to annotate available corpora. At the end of this chapter, we enumerate the tags of word forms in the given sentence used in all designed Czech tag sets (Table 3.8).

3.1.1 Czech Corpus

Thanks to the enthusiasm of a group of researchers from the Institute of Czech Language headed by M. Těšitelová the main corpus of written and spoken Czech was created during the 60s and 70s. The main motivation for building CC was to obtain the quantitative characteristics of present-day Czech. The corpus includes newspapers, journals, scientific and literary texts. The quantitative research concentrated among other things on the frequency of part of speech classes, frequency of morphological categories and of certain syntactic phenomena. For these purposes CC was morphologically and syntactically annotated. Tags used in CC were different from our suggested tags (CTC tag system, see Tables 3.1 and 3.2) especially as concerns the number of processed MCs and the notation. Thus we carried out conversions of the original data (CC) into the Czech Tagged Corpus (CTC) CTC tag system. As we are interested how the tagging accuracy changes as the amount of information included in tags changes, we design reduced CTC tag system (RCTC tag system) and map CTC into Reduced Czech tagged corpus (RCTC).

3.1.2 Czech Tagged Corpus

As mentioned above, CC was originally morphologically annotated, including lemmatization and syntactic tags. For the purpose of Czech tagging experiments, we have used only a part of the CC and we have disregarded the lemmatization information and the syntactic tags, as we were interested in word tokens and tags only.

CTC Tag Set

The CTC tag set was derived from the original tag set used to annotate the CC. CTC tag set is designed ([Hladká, 1994]) in a similar way as the tag systems traditionally used for English: the first letter of the tag defines the part of speech (POS) class and the remaining letters express the values of morphological categories within a particular POS. Table 3.1 provides a complete list of the MCs (together with their variables and all possible values, totally nine MCs) which we are interested in. We concentrate on 10 major part of speech classes - *nouns (N), adjectives (A), verbs (V), pronouns (P), numerals (C), adverbs (D), conjunctions (C), prepositions (R), particles (K), interjections (F)*. In addition to these typical POS classes we used three specific classes, namely *punctuation marks (T_IP), sentence boundaries (T_SB), unknowns (X)*. However, not all MCs are involved in the inflection of each POS class. The meaningful sequences of the MCs in the pre-specified order for 10+3 POS classes are defined in Table 3.2. Some of the major POS classes are associated with detailed POS category (so called sub-POS category), i.e the tag for the given POS class contains not only the sequence of the MCs but also the letter identifying the sub-POS category.

For instance, the MCs `gender, number, case` are involved in the inflection of nouns; according to the pre-specified order of MCs within the noun tags, we can easily decode the tag NFS3: feminine (F - the 2nd position) singular (S - the 3rd position) noun (N - the 1st position) in dative (3 - the 4th position).

In our CTC tag system, the POS class pronouns (P) represents the category associated with the most detailed division into sub-POS categories - personal pronouns (PP), possessive pronouns (PR), "svůj" reflexive (posesssive) pronoun (PS), reflexive particle (PE) and demonstrative pronouns (PD). For example, the pronoun "svůj" is characterized by MCs object `gender`, object `number, case` ($\mathrm{PS}g_2n_2c$).

The prepositional (R) tags do not include a sequence of MCs, they just consist in the given preposition, i.e. the prepositional tag "Rpřed" represents the tag of the preposition "před"("in front of").

There is no MC involved in the inflection of interjections (F), particles (K), sentence boundaries (T_SB), punctuation (T_IP) and unknowns (X).

3.1.3 Reduced Czech Tagged Corpus

Once the annotated corpus is available, we can get several annotated corpora which differ from the original one in the reduction of the original corpus tag set. In other words, we map the original tag set into a less detailed tag set

Morphological Category	Category Variable	Possible Value	Description
gender	g,(g_1, g_2)	M	masculine animate
		I	masculine inanimate
		N	neuter
		F	feminine
number	n, n_1, n_2	S	singular
		P	plural
tense	t	M	past
		P	present
		F	future
mood	m	O	indicative
		R	imperative
case	c	1	nominative
		2	genitive
		3	dative
		4	accusative
		5	vocative
		6	locative
		7	instrumental
voice	s	A	active voice
		P	passive voice
polarity	a	N	negative
		A	affirmative
degrees of comparison	d	1	base form
		2	comparative
		3	superlative
person	p	1	1st (I, we)
		2	2nd (you)
		3	3rd (she, he, it, they)
	f	1	1st (I, we)
		2	2nd (you)

Table 3.1: Morphological categories included in (R)CTC tag set

POS CLASS	CTC TAG	RCTC TAG
nouns	Ngnc	Nn
noun, abbreviations	NZ	NZ
adjectives	A$gncda$	Anda
verbs, infinitives	VTa	VTa
verbs, transgressives	VW$ntsga$	VW$ntsa$
verbs, common	V$pnstmga$	V$pnsta$
pronouns, personal	PPfnc	PPfn
pronouns, 3rd person	PP3gnc	PP3n
pronouns, possessive	PR$g_1n_1cpg_2n_2$[a]	PRn_1pn_2
"svůj" — reflexive possessive pronoun	PSg_2n_2c	PSn_2
reflexive particle "se"	PEc	PE
pronouns, demonstrative	PD$gnca$	PDna
adverbs	Oda	Oda
conjunctions	S[S|P]	S[S|P]
numerals	Cgnc	Cn
prepositions	Rpreposition	Rpreposition
interjections	F	F
particles	K	K
sentence boundaries	T_SB	T_SB
punctuation	T_IP	T_IP
unknown tag	X	X

[a] g_1/n_1 - possessor's gender/number, g_2/n_2 - object gender/number with possessive pronouns

Table 3.2: CTC tag set and RCTC tag set

FEATURE	
# of word tokens	622K
# of different tags	1,171
average number of tags per word form	3.65

Table 3.3: CTC characteristics

so that we disregard the chosen MCs. In practice, we have decided to use CTC in order to get RCTC by CTC tag set reduction.

Reduced CTC Tag Set

The RCTC tag set is based on the same MCs (see Tab. 3.1) as the CTC tag set except the MCs `case`, `gender` and `mood`. Table 3.2 illustrates the decrease (expressed by the RCTC tag set) in the amount of information included into CTC tag set.

FEATURE	
# of word tokens	622K
# of different tags	206
average number of tags per word form	2.23

Table 3.4: RCTC characteristics

3.1.4 Czech National Corpus

The Czech National Corpus (CNC) is being built-up by an concerted effort of a number of institutions, mostly by the Institute of Czech National Corpus, Faculty of Philosophy, Charles University. The work on CNC has started at the beginning of the 90s. Primarily, the CNC (as a representative collection of Czech synchronic and diachronic texts obviously stored on computer) is supposed to serve as a source for the build up of a dictionary of contemporary Czech. At this time, the number of word form tokens included reaches 100 million.

3.1.5 Prague Dependency Treebank

The work on the Prague Dependency Treebank (PDT) started in 1997 and is still "under construction". The PDT is conceived of as a corpus of Czech texts with a rich annotation scheme. PDT has a three level structure: morphological ([Hajič, Hladká, 1998a]), analytical syntactic level ([Hajič, Hajičová, Panevová, Sgall, 1998], [Bémová et al., 1999]) and level of linguistic meaning ([Hajičová, 1998], [Hajičová, Panevová, Sgall, 1998], [Böhmová, Hajičová, 1999], [Böhmová, Panevová, Sgall, 1999]).

Textual data for the PDT are selected from CNC and are pre-processed by the automatic morphological analysis (AMA) that gives a list of all possible positional tags (see below) for the given word forms. Currently, the PDT

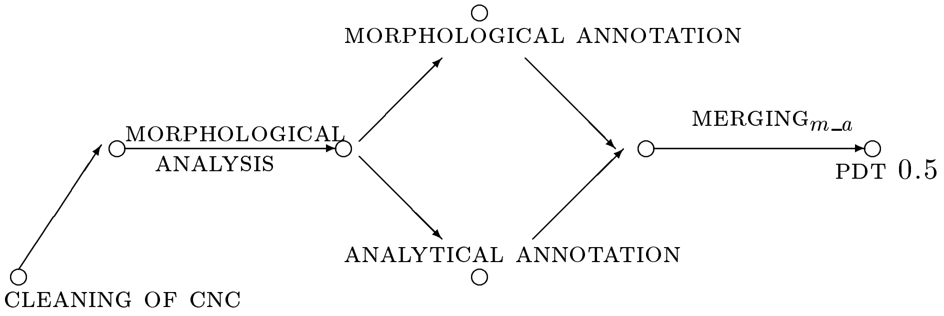

Figure 3.1: The scheme of building PDT version 0.5

version 0.5 (26,610 sentences, 456,705 word tokens) is available. The PDT 0.5 has been prepared in the way displayed in Fig. 3.1: texts from CNC are cleaned (removing of the duplicates of documents, paragraphs, sentences, spelling errors checking, etc.), morphologically and analytically annotated; because of the chosen strategy of annotations on different levels, it is necessary to merge (procedure MERGING$_{m_a}$) the annotations into a single data resource. As the cleaning of CNC texts and the annotations are executed in a parallel way, the merging process cannot be fully automatic because of the different text versions (in the sense of cleaning phases) which enter the annotations. A semi-automatic procedure is applied and the discrepancies must be resolved manually.

Positional Tag System

Czech positional tags are defined as a concatenation of 15 MCs[1] which are introduced in Table 3.5; each MC corresponds to precisely one position so that `POS` "sits" in the 1st position, `SubPOS` in the 2nd, `g` in the 3rd, `n` in the 4th, `c` in 5th, `possg` in the 6th, `possn` in the 7th, `p` in the 8th, `t` in the 9th , `d` in the 10th, `a` in the 11th, `v` in the 12th and `s` in the 15th position[2]. We present only the MCs we work with in the positional tag system and we do not deal with the possible values of MCs here. A very detailed description of the positional tag system of Czech is presented in [Hajič, in press].

[1]Actually, there are 13 MCs currently used for Czech and there are two more categories `x1`, `x2` the values of which are not defined yet. In other words, the sets of all possible values **RESERVE1**, **RESERVE2** are empty.

[2]Positions 13 and 14 correspond to MCs `x1` and `x2`.

Category	Variable ∈ Set of category values	
part of speech	`POS`	∈ **POS**
sub-part of speech	`SubPOS`	∈ **SUBPOS**
gender	`g`	∈ **GENDER**
number	`n`	∈ **NUMBER**
case	`c`	∈ **CASE**
possessive gender	`possg`	∈ **POSSGENDER**
possessive number	`possn`	∈ **POSSNUMBER**
person	`p`	∈ **PERSON**
tense	`t`	∈ **TENSE**
degree	`d`	∈ **DEGREE**
negation	`a`	∈ **NEGATION**
voice	`v`	∈ **VOICE**
not defined	`x1`	∈ **RESERVE1**
not defined	`x2`	∈ **RESERVE2**
variation	`s`	∈ **VAR**

Table 3.5: Individual morphological categories and their variables

Formally, let CZT be a set of all theoretically possible Czech positional tags and Cz_T be a set of all meaningful Czech positional tags: $CZT = POS \times SUBPOS \times GENDER \times NUMBER \times CASE \times POSSGENDER \times POSSNUMBER \times PERSON \times TENSE \times DEGREE \times NEGATION \times VOICE \times RESERVE1 \times RESERVE2 \times VAR, Cz\_T \subset CZT$.

3.1.6 Xerox Czech Tagged Data

[Schiller, 1996] describes the general architecture of the Xerox Language Tool (XLT) for noun phrase mark-up and a statistical tagger for seven European languages based on finite state techniques. The XLT tagger (belonging to the MM category; [Cutting at al., 1992], [Tapanainen, 1995]) does not require annotated data (or very small amount of them) and uses the so called Baum-Welch algorithm ([Baum, 1972]) for the generation of a HMM. To train the Xerox tagger on Czech, we annotated 15,000 word tokens long newspaper texts selected from CNC. As described below, we designed three tag sets for the purposes of the Xerox experiments. Consequently, we annotated the specified texts by these tag sets and we obtained the annotated corpora $\mathrm{CNC}^{47}_{Xerox}$ (Xerox_{47} tag set), $\mathrm{CNC}^{43}_{Xerox}$ (Xerox_{43} tag set) and $\mathrm{CNC}^{34}_{Xerox}$ (Xerox_{34} tag set).

Xerox Tag Sets

We performed three Xerox experiments, which differ in the tag set. The analysis of the results of the first experiment (Xerox_{47} tag set) showed a very high ambiguity between the nominative and the accusative `case` of nouns, adjectives, pronouns and numerals. This fact was a strong motivation to replace the tags for nominative and accusative of nouns, adjectives, pronouns and numerals by new tags NOUN_NA, ADJ_NA, PRON_NA and NUM_NA (meaning nominative or accusative, undistinguished). The rest of tags stayed unchanged. This modification resulted in the Xerox_{43} tag set. In the next step, we deleted the morphological information (excluding `part of speech` information) for nouns and adjectives altogether. This process resulted in the final Xerox_{34} tag set. Table 3.6 provides a complete list of morphological categories we work with during the Xerox experiments and Table 3.7 presents the tags for the given POS classes.

3.1.7 Annotation Using the Six Different Czech Tag Systems

In the course of the creation of annotated language resources, there have been designed six different Czech tag sets in total. The MCs in the CTC tag set are same as the MCs assumed in the Czech positional tag set except for the MCs `sub-part of speech` and `variation`. However, the list of possible MC values in the positional tag system is greater than in the CTC tag set. We demonstrate the influence of the tag set size on the tagging accuracy on the example of the RCTC tag set and XEROX tag sets that are not so detailed as the CTC and the positional tag sets.

To give an example of the annotation using the different tag systems, we have chosen an annotated sample sentence from the CTC:

Knihkupec	Václav	Klement	a	strojník	Václav	Laurin	se	dohodli	na společné
Bookseller	Václav	Klement	and	mechanic	Václav	Laurin	Refl.	have agreed	on joint
výrobě	jízdních	kol.							
production	of-bicycles.								

'The bookseller Václav Klement and the mechanic Václav Laurin have agreed on joint production of bicycles.'

Table 3.8 contains not only CTC tags but also the appropriate tags from the other Czech tag sets.

MORPHOLOGICAL CATEGORY	CATEGORY VARIABLE	POSSIBLE VALUE	DESCRIPTION
case	c	NOM	nominative
		GEN	genitive
		DAT	dative
		ACC	accusative
		VOC	vocative
		LOC	locative
		INS	instrumental
case	c'	NA	nominative or accusative
		GEN	genitive
		DAT	dative
		VOC	vocative
		LOC	locative
		INS	instrumental
kind of verb form	k	PAP	past participle
		PRI	present participle
		INF	infinitive
		IMP	imperative
		TRA	transgressive

Table 3.6: Morphological categories included in Xerox tag sets

POS CLASS	XEROX$_{47}$ TAG SET	XEROX$_{43}$ TAG SET	XEROX$_{34}$ TAG SET
nouns	NOUN_c	NOUN_c'	NOUN
abbreviations	NOUN_INV	NOUN_INV	NOUN
adjectives	ADJ_c	ADJ_c'	ADJ
verbs	VERB_k	VERB_k	VERB_k
pronouns	PRON_c	PRON_c'	PRON_c
reflexive particle “se”	P_SE	P_SE	P_SE
adverbs	ADV	ADV	ADV
conjunctions	CONJ	CONJ	CONJ
numerals	NUM_c	NUM_c'	NUM_c
numbers	NUM_INV	NUM_INV	NUM_INV
prepositions	PREP	PREP	PREP
interjections	INTJ	INTJ	INTJ
particles	PTCL	PTCL	PTCL
sentence boundaries	SENT	SENT	SENT
punctuation (excluding comma)	PUNCT	PUNCT	PUNCT
comma	CM	CM	CM
proper names	PROP	PROP	PROP
clitics	CLIT	CLIT	CLIT
date	DATE	DATE	DATE

Table 3.7: Xerox tag sets

TAG SYSTEM	CTC	RCTC	POSITIONAL	XEROX$_{47}$	XEROX$_{43}$	XEROX$_{34}$
Knihkupec	NMS1	NS	NNMS1-----A----	NOUN_NOM	NOUN_NA	NOUN
Václav	NMS1	NS	NNMS1-----A----	NOUN_NOM	NOUN_NA	NOUN
Klement	NMS1	NS	NNMS1-----A----	NOUN_NOM	NOUN_NA	NOUN
a	SS	SS	J^-------------	CONJ	CONJ	CONJ
strojník	NMS1	NS	NNMS1-----A----	NOUN_NOM	NOUN_NA	NOUN
Václav	NMS1	NS	NNMS1-----A----	NOUN_NOM	NOUN_NA	NOUN
Laurin	NMS1	NS	NNMS1-----A----	NOUN_NOM	NOUN_NA	NOUN
se	X	X	P7-X4----------	P_SE	P_SE	P_SE
dohodli	V3PAMOMA	V3PAMA	VpMP---3R-AA--1	VERB_PAP	VERB_PAP	VERB_PAP
na	Rna	Rna	RR--6----------	PREP	PREP	PREP
společné	AFS61A	AS1A	AAFS6----1A----	ADJ_LOC	ADJ_LOC	ADJ
výrobě	NFS6	NS	NNFS6-----A----	NOUN_LOC	NOUN_LOC	NOUN
jízdních	ANP21A	AP1A	AANP2----1A----	ADJ_GEN	ADJ_GEN	ADJ
kol	NIP2	NP	NNNP2-----A----	NOUN_GEN	NOUN_GEN	NOUN

Table 3.8: The example of annotation of the sentence using the different Czech tag sets

3.1.8 The Penn Treebank

A very detailed review of the experience from building the Penn Treebank is given in [Marcus, Santorini, Marcinkiewicz, 1993]. Wall Street Journal (WSJ) is the main annotated and parsed part (1.2 Mw) of the Penn Treebank.

The Penn Treebank tag set

The Penn Treebank tag set[3] is given in Appendix A. It is designed for English and contains 36 tags and 12 other tags (for punctuation and currency symbol).

3.2 Is Tagging of Czech Different from Tagging of English?

At the first sight we would say *yes*, the tagging of Czech should be different from the English one. The answer can be found in the cardinality of the English tag set and Czech tag sets; for English, we work with the Penn Treebank tag set. The difference between a morphologically complex and ambiguous inflective language and a language with poor inflection is reflected, e.g., in the number of tags for adjectives. Table 3.9 provides factual numbers.

Penn Treebank tag set distinguishes three tags for adjectives - JJ (adjective), JJR (adjective comparative) and JJS (adjective superlative). For Czech, we illustrate the differences between Czech and English using four Czech tag sets: CTC, RCTC, positional and Xerox. We get the number of all possible adjective tags as the product of the number of possible values of those MCs which are involved in the adjective inflection within the given tag set. As the positional tag set is the most detailed (namely in the course of sub-POS categories), the number of possible adjective positional tags outnumbers the others. The last column of Table 3.9 provides conclusive empirical evidence (3 vs. 336; 12; 7; 6; 1; 2,916) of the morphological difference between Czech and English.

In addition to the morphological ambiguity it is also interesting to note the frequencies of the most ambiguous word forms encountered in the whole CTC and to compare them with the English data. Table 3.10 and Table 3.11 contain the first word forms with the highest number of possible tags in the complete CTC and in the complete WSJ.

[3]Sect. 3.2 provides a detailed analysis of the Penn Trebank tags.

TAG SET	ADJECTIVE TAGS		# OF ADJECTIVE TAGS
Penn Treebank	JJ, JJR, JJS		**3**
CTC	A[MIFN][SP][1234567][123][AN]	1x4x2x7x3x2 =	**336**
RCTC	A[SP][123][AN]	1x2x3x2 =	**12**
$Xerox_{47}$	ADJ_[NOM|GEN|DAT|ACC|VOC|LOC|INS]	1x7 =	**7**
$Xerox_{43}$	ADJ_[NA|GEN|DAT|VOC|LOC|INS]	1 x 6 =	**6**
$Xerox_{34}$	ADJ		**1**
Positional	adjective general AA[FIMN][DPS][1234567]----[123][AN]---[-678]	1x1x4x3x7x3x2x4 =	**2,016**
	adjective nominal AC[FMNQT][PSW][-4]-----[AN]----	1x1x5x3x2x2 =	**60**
	adjective derived from present transgressive form of a verb AG[FMIN][DPS][1234567]-----[AN]---[-67]	1x1x4x3x7x2x2 =	**336**
	adjective derived from verbal past transgressive form AM[FMIN][DPS][1234567]-----[AN]---[-67]	1x1x4x3x7x2x2 =	**336**
	adjective possessive AU[FMIN][DPS][1234567][FM]---------	1x1x4x3x7x2 =	**168**
	TOTAL POSITIONAL ADJECTIVE TAGS		**2,916**

Table 3.9: Czech tag sets vs. Penn Treebank tag set

WORD FORM	FREQUENCY IN CTC	# OF TAGS IN CTC
jejich	1,087	51
jeho	1,087	46
jehož	163	35
jejichž	150	25
vedoucí	193	22

Table 3.10: The most ambiguous word forms in CTC

WORD FORM	FREQUENCY IN WSJ	# OF TAGS IN WSJ
a	25,791	7
down	1,052	7
put	380	6
set	362	6
that	10,902	6
the	56,265	6

Table 3.11: The most ambiguous word forms in WSJ

2	AFP11A	2	ANP11A
4	AFP41A	2	ANS41A
6	AFS11A	10	NFS1
11	AFS21A	1	NFS2
1	AFS31A	1	NFS3
4	AFS41A	1	NFS4
5	AFS71A	2	NFS7
2	AIP11A	34	NMP1
11	AMP11A	17	NMP4
3	AMP41A	61	NMS1
12	AMS11A	1	NMS5

Table 3.12: Tags of “vedoucí” in CTC

To go back to the morphological ambiguity of adjectives, for instance in the CTC, the word form ”vedoucí” appeared 193 times and was annotated by twenty two different tags: 13 tags for adjective and 9 tags for noun. The word form ”vedoucí” means either: ”leading” (adjective) or ”manager”, ”boss” (noun). The columns in Table 3.12 represent the tags for the word form ”vedoucí” and their frequencies in the CTC; for example ”vedoucí” was annotated twice as adjective, feminine, plural, in nominative, first degree, affirmative (2 AFP11A).

It is an obvious fact that there exists a strong relationship between the tag set and the tag bi/trigrams in the annotated corpus: the more detailed tag set, the higher is the number of possible tag bigrams and tag trigrams. Theoretically, let $TAGS$ be a tag set. Thus $|TAGS|^2$ is the number of all possible tag bigrams and $|TAGS|^3$ is the number of all possible tag trigrams. Be $|TAGS| = 10^2$, then $|TAGS|^2 = 10^4$ and $|TAGS|^3 = 10^6$; consequently, there must be many tag bigrams and tag trigrams that are infrequent; Tables 3.13 and 3.14 illustrate this for CTC and WSJ. With regard to the number of different tags in CTC and WSJ (1,171 vs. 48), the numbers of different tag bigrams (33,928 vs. 1,453) and tag trigrams (177,083 vs. 18,257) in the CTC and the WSJ are completely different. In the CTC, 70.93% of tag bigrams appear less than four times and 4,66% of tag bigrams appear more than sixteen times. In the WSJ, to get reasonably high counts, we do not use such low limits; 31.59% of tag bigrams appear less than ten times and 15.49% of tag bigrams appear more than one thousand times. For tag trigrams, the situation is similar. We would like to stress that the CTC has a total length of 622K word tokens and the WSJ is 1.2M word tokens long. It seems that we cannot directly compare the counts of tag bigrams and tag trigrams due to the different size of corpora. In spite of this fact, we are sure that since we apply tagging methods based on lexical and tag context, the more detailed tag set of Czech we use the more different is tagging of Czech from tagging of English.

It is clear from these observations that due to the fact that the two languages in question have quite different properties; nothing can be said about the possible tagging differences without really going through an experiment.

	CTC		WSJ
x<=4	24,064	x<=10	459
4<x<=16	5,577	10<x<=100	411
16<x<=64	2,706	100<x<=1,000	358
x>64	1,581	x>1,000	225
total # of tag bigrams	33,928	total # of tag bigrams	1,453

Table 3.13: Number of tag bigrams with frequency x in CTC and WSJ

3.3 Czech Automatic Morphological Analysis

In [Hajič, in press], the author deals with the computational Czech morphology and tagging (language independent) and the description of their mutual relationship. Using the mathematical terms, the author defines morphological analysis (MA) as a function MA: $F \to 2^{LxT}$:

MA(f) = {<l,t>, l $\in$ L & t $\in$ T} $\subseteq$ LxT, f $\in$ F,

	CTC		WSJ
x<=4	155,399	x<=10	11,810
4<x<=16	16,371	10<x<=100	4,571
16<x<=64	4,380	100<x<=1,000	1,645
x>64	933	x>1,000	231
total # of tag trigrams	177,083	total # of tag trigrams	18,257

Table 3.14: Number of tag trigrams with frequency x in CTC and WSJ

where F is a set of word forms, L is a set of lemmas, T is a set of tags. Given the tokenized input text, the MA gives all possible morphological analyses of the word forms. Then, tagging is a function $\Phi : F \rightarrow T$: Φ(f) = t so that $\exists$ l $\in$ L: <l,t>$\in$ MA(f). At this point, we have to stress that "Czech" tagging strategies which operate over the morphologically preprocessed text take into account no lemma information. To tag text in the form word token w, lemma l, tag t we (i) choose tag t according to the tagging algorithm and then (ii) take such lemma l that $< l, t > \in MA(w)$.

3.4 Tagging Experiments

In this section, we present results of various corpus-based techniques applied to tag Czech texts in order to show how these techniques work for one of the highly morphologically ambiguous inflective languages. Together with the description of concrete variants of tagging techniques we concentrate on a detailed analysis of the results. Table 3.29 provides a complete overview of all accomplished experiments.

3.4.1 MM Strategy

We have used the basic Markov models described in Sect. 2.3. Our implementation is driven by the Eqs. 2.13 (bigram MM) and 2.14 (trigram MM) expressing the conditions put on an optimal tag sequence given the input text.

The first code (under MS-DOS platform) that carries out the Czech tagging via bigram MM, Viterbi algorithm and an "intuitive"[4] smoothing procedure appeared in the course of the work described in [Hladká, 1994]. As the next ideas came up through the series of the experiments, we enriched

[4]We tune the λ parameters using an empirical experience.

the given code; we processed not only bigram MM but trigram MM as well, and we included the output of MA. The latest versions of the code are also running under Unix platform.

Altogether, we have performed two sets of the MM experiments: the first one *excluding* morphological preprocessing and the second one *including* morphological preprocessing. The difference is determined by the way in which the set TAGS_w is generated (the set of all meaningful tags for a given word token w is obtained, see Sect. 2.1). The experiment without morphological preprocessing does not cooperate with any morphological analyser to get the sets TAGS_w; thus the set TAGS_w for each word token w in a file to be tagged contains all different tags which occur in the training corpus. In this case, it is too audacious to speak about the set of meaningful tags for the given word token w. On the other hand, the experiment with morphological preprocessing works with really meaningful tags for the given word token obtained via morphological analyser.

Later on, the new version of Czech MM tagger which tags a text that is morphologically preprocessed has appeared ([Mírovský, 1998]). This version operates only with contextual probabilities (no lexical probabilities; Sect. 2.3). Then the condition put on the optimal tag sequence (Γ) has the following form:

$$\Gamma \approx \max_{T} p(T) \tag{3.1}$$

We have already discussed the contextual tag probabilities, and the equations 2.16, 2.20 (unigram MM), 2.17, 2.21 (bigram MM) and 2.18, 2.22 (trigram MM) are valid for the model covering only contextual information; the smoothing procedure is realised via EM-algorithm ([Jelinek, Mercer, 1980]). The result for Czech is presented in Tab. 3.29 (experiment Mannheim).

Without Morphological Pre-Processing

We have split the complete CTC into two mutually exclusive parts: the bigger part (621,015 word tokens) was used as a training file ($\mathrm{CTC}^{621}_{train}$) and the smaller part (1,294 word tokens) as a test file (CTC_{test}). Identically to the splitting of CTC, we have split RCTC into the parts $\mathrm{RCTC}^{621}_{train}$, RCTC_{test} directly corresponding to the parts $\mathrm{CTC}^{621}_{train}$, CTC_{test}. Even more, we have separated from $\mathrm{CTC}^{621}_{train}$ the file $\mathrm{CTC}^{110}_{train}$ containing only 110K word tokens. The six experiments based on the MMs differ in three aspects: (i) the order of MM, (ii) the training data size and (iii) tag set size; we will discuss the performance of the experiment according to these three aspects.

Tab. 3.15 lists the experiment parameters. We will discuss the results and the aspects of the experiments using as background Fig. 3.2.

EXPERIMENT ID	MM MODEL	CORPUS	TRAINING DATA SIZE
Hlinsko	unigram	CTC	621,015
Prague	bigram	CTC	621,015
Mariánská	trigram	CTC	621,015
Copenhagen	bigram	CTC	110,874
Granada	bigram	RCTC	621,015
London	trigram	RCTC	621,015

Table 3.15: The specification of the MM experiments without morphological preprocessing

order of MM — Involvement of the tag history in the tag prediction gives significantly better results than a simple assignment of the most probable tag (unigram MM vs. bigram MM, unigram MM vs. trigram MM). Comparing the results of bigram and trigram MMs, we cannot conclude that including the tags of two previous word tokens gives better results than including only the tag of the preceding word token because of the simple fact that 80.91% $\not>$ 81.38%. In Tabs. 3.13 and 3.14 we presented the idea concerning the number of different bigram tag histories and trigram tag histories. Clearly, if 88% trigrams occur four times or less, then the statistics are not reliable. In order to get better results for a trigram prediction model, we would need a much larger amount of data.

training data size — The experiments Prague and Copenhagen show (not surprisingly) that the more training data, the better the success rate.

tag set size — In [Elworthy, 1995] experiments with changing tag sets are presented for three different languages (English, French, Swedish). These experiments show that the relationship between tag set size and accuracy is a weak one and is not consistent even if applied for the same language. The main conclusion derived from the results of experiments is to choose the tag set according to the requirements of the given post-tagging application rather than to optimize it for the tagger. The question arises why it is so useful to add a morphological information to a text. The adequate answer is that there are applications for which tagged input data are needed and it is an obvious fact that these applications presuppose a text tagged as thoroughly as possible. We have decided to reduce the CTC tag set not only because of the specific mutual dependence between the size of the tag set and the tagging accuracy but because of the relation of tag set and the results of the parsing procedure ([Ribarov, 1996]) as well. The deduction presented in [Hladká, Ribarov,1998] says that a reduced tag set ($Xerox_{34}$) brings better absolute success values. On the other hand, it could seem very strange to disregard such important MCs for Czech as `case` and `gender` (CTC tag set → RCTC tag set). Immediately, the tagging accuracy increases from 81.38% (experiment Prague) to 90.11% (experiment Granada) and from 80.91% (experiment Mariánská) to 90.30% (experiment London); the relatively high performance is achieved at the cost of the omitted morphological information which could be so important for a number of post-tagging applications.

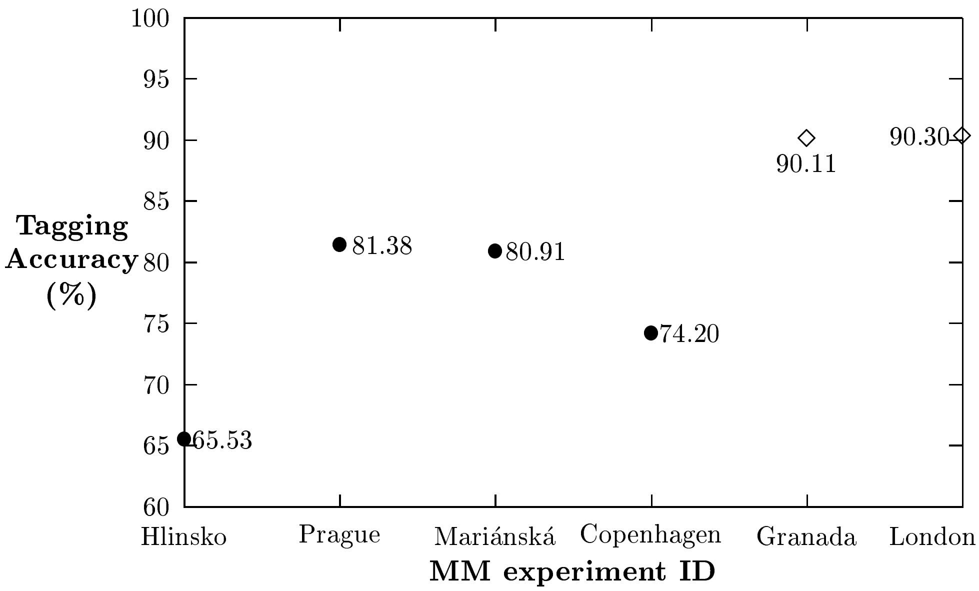

Figure 3.2: The results of the MM experiments without morphological preprocessing

The way of defining the tagging accuracy (Eq. 2.1) concerns only the tag level perspective. It says nothing about the subtag level errors. In order to know what kind of errors the MM taggers produce we analysed the output of trigram MM tagger trained on $\mathrm{CTC}_{train}^{621}$ (experiment Mariánská).

The letters in the first column and the corresponding rows of Tab. 3.16 denote 10 basic POS classes + 2 additional classes (see Par. 3.1.2) (for the evaluation, the punctuation class and sentence boundary class are involved under a single class marked by T). The numbers show how many times the tagger assigned an incorrect tag to a word token in the test file. The total number of errors was 244. Altogether, the adjectives (A) were tagged incorrectly fifty times, nouns (N) 93 times, numerals (C) 5 times, etc. (see the last unmarked column); to provide a better insight, we should add that in 32 cases out of 50, the adjective was correctly tagged as an adjective, but the mistakes appeared in the assignment of the other morphological categories.

	A	C	F	K	N	O	P	R	S	T	V	X	
A	32	0	0	0	6	3	2	2	2	1	2	0	50
C	0	4	0	0	1	0	0	0	0	0	0	0	5
F	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	1	0	0	1	2
N	4	0	0	0	64	8	0	4	2	5	2	4	93
O	0	0	0	0	1	0	0	0	0	1	1	0	3
P	0	0	0	0	0	3	19	0	0	0	0	1	23
R	0	0	0	0	1	1	0	0	0	0	0	2	4
S	0	0	0	0	0	0	0	0	0	0	0	2	2
T	0	0	0	0	1	0	0	0	0	0	0	0	1
V	0	0	0	0	3	8	0	3	8	1	28	2	53
X	0	0	0	0	0	0	5	0	1	0	2	0	8

Table 3.16: The distribution of the errors produced in the trigram experiment on CTC

Tables 3.17-3.20 provide a very concrete idea about the kind of the incorrect assignments of MCs in case of a correct part of speech class assignment. Going back to the adjectives (Tab. 3.17), only `gender` of the 17 adjectives was tagged incorrectly, only `number` was wrong once, six times `case` was an error, `gender` and `case` were mistagged three times jointly, etc. Putting together the numbers in the error evaluation tables, the numbers confirm the expected fact that MCs `case`, `gender` and `number` belong to the most errorable Czech MCs.

A	g	n	c	g&c	g&n	c&a	g&n&c	g&c&d
32	17	1	6	3	2	1	1	1

Table 3.17: The distribution of the adjective errors produced in the trigram experiment on CTC

N	g	n	c	g&c	n&c	$\rightarrow$ NZ
64	11	5	41	2	4	1

Table 3.18: The distribution of the noun errors produced in the trigram experiment on CTC

C	g	c	P	g	c	g&c	PD $\rightarrow$ PP
4	1	3	19	8	7	3	1

Table 3.19: The distribution of the numeral and pronoun errors produced in the trigram experiment on CTC

For running the experiments on English (without any morphological pre-processing), we had to change the format of WSJ to prepare the data for our MM tagging software. The numbers in Tab. 3.21 confirm our corpus-driven (not experiment-based!) assumption concerning the differences in tagging of Czech and of English; the results for English outperform results for Czech significantly.

With Morphological Pre-Processing

Since the start of the tagging experiments it has been clear that including linguistic information into purely statistical approaches should be a step forward. The term linguistic information means (in our case) linguistic information obtained from the MA, although we are aware that also more general layers of linguistic information are relevant. As mentioned in Sect. 2.1, we need a set of meaningful tags TAGS_w for a given word token w. Altogether, the linguistic information received from the MA is expressed in the TAGS_w for each word token w in the text.

V	p	t	n	s	n&t	p&t	t&a	g&a	p&n&t	V → VT
28	3	6	5	5	1	1	1	1	1	4

Table 3.20: The distribution of the verb errors produced in the trigram experiment on CTC

STRATEGY	CORPUS	TRAINING DATA SIZE	TAGGING ACCURACY (%)
unigram MM	WSJ	1,287,749	89.50
bigram MM	WSJ	1,287,749	96.38
trigram MM	WSJ	1,287,749	97.14
bigram MM	WSJ	110,530	93.74

Table 3.21: The experiments on English using the MM strategy

The specification of the experiments with the morphological preprocessing are listed in Tab. 3.22. The power of linguistic information if included in the pure statistical method can be illustrated by Fig. 3.3.

linguistic information — Having the training data of a comparable size (experiment Copenhagen vs. experiments Birmingham, Montecatini), a comparable number of different tags (882 vs. 860) and having the same tagging strategy (MM) allows us to make a fair comparison of taggers without and with morphological preprocessing. The presence of the list of all meaningful tags for each word token in the test file causes an improvement from 74.2% to 87.8% in TA of the tagging procedure. The demand for more training data in case of a tagger with morphological preprocessing becomes more intensive.

EXPERIMENT ID	MM MODEL	CORPUS	TRAINING DATA SIZE
Birmingham	bigram	PDT	124,692
Montecatini	trigram	PDT	124,692

Table 3.22: The specification of the MM experiments with morphological preprocessing

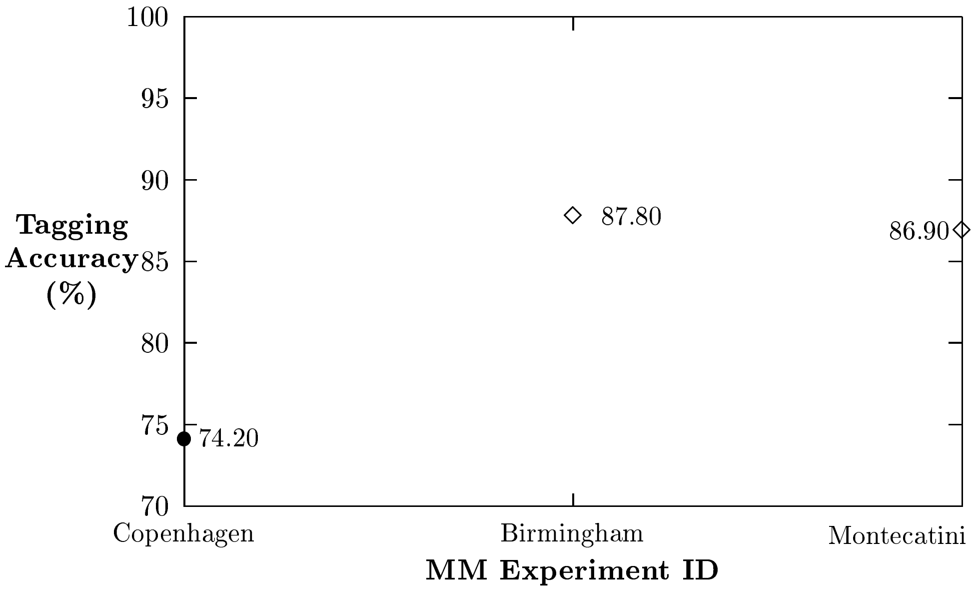

Figure 3.3: The comparison of the MM experiments with/without morphological preprocessing

3.4.2 RB Strategy

For Czech, we take the rule-based tagger "as it is" (designed for English), i.e. with the prespecified lexical/contextual templates of the following form ([Brill, 1993a]):

lexical templates

Change the most likely tag to X

- if deleting the prefix x, $|x| \leq 4$, results in a word form
- if the first (1,2,3,4) characters of the word form are x
- if deleting the suffix x, $|x| \leq 4$, results in a word form
- if the last (1,2,3,4) characters of the word form are x
- if adding the character string x as a suffix results in a word form ($|x| \leq 4$)
- if adding the character string x as a prefix results in a word form ($|x| \leq 4$)

contextual templates

Change the most likely tag to X

EXPERIMENT ID	CORPUS	TRAINING DATA SIZE
Baltimore	CTC	75,863
Washington	CTC	19,565

Table 3.23: The specification of the RB experiments

- if the word token w ever appears immediately to the 1/2/3 positions left/right of the word token
- if the word token w is the current word token
- if the word token w tagged by tag t appears immediately left/right of the word token
- if the word bigram $w_1,\ w_2$ appears immediately left/right of the word token
- if the tag t ever appears immediately to the 1/2/3 positions left/right of the word token

The strategy of a rule-based tagger determines the usage of annotated and unannotated corpora. The annotated corpus is being split into two parts of equal size. The first of these parts is used for learning the rules to predict the most probable tag for unknown words (lexical rules) and the second one is used for learning contextual rules.

For Czech, we used the complete CTC with tags removed as an unannotated corpus. Two experiments differ in the annotated corpus (part of CTC) size; see Tab. 3.23. Needed training time is the reason we separated relatively small parts with regard to the size of the complete CTC is in the time needed for training.

The disadvantage of statistical methods is the "opaqueness" of the computational process across the large statistical tables. At the same time, in contrast to the large statistical tables, it is a pleasure to follow a small set of rules generated during the learning phase of a rule-based tagger.

The lexical rules learn the morphology from the training data by "playing" with suffixes and prefixes of word forms. The lexical templates (as they are designed for English) look at up to the four first or last characters in a word form. It seems that the number four is suitable for Czech as well.

Table 3.24 offers those lexical rules which belong to the lexical rules generated during the training step of experiments Baltimore and Washington at the beginning of training.

Tab. 3.25 provides the word token samples which satisfy the conditions determined by the rules listed in Tab. 3.24. For instance, the rule 25 (Tab. 3.24) says: if the given word form has the two letter suffix "ým", then change the tag of the word token to the tag AIS71A. Since the word token "novým" satisfies the left-hand side of the given rule, we can change the tag of "novým" to the tag AIS71A. The usage of an unannotated corpus is obvious from the rules No. 11, 23 and 24 which add the suffixes to the given word form and test if the result is a word (in an unannotated corpus). Let "problém" be the currently processed word form; adding one-letter suffix "u" results in the word "problému". If the word form "problému" appears in unannotated corpus, then the tag of "problém" is changed to the tag NIS1 (see rule No. 11).

	if			Change to		if			Change to
1	t	char[a]		VTA	14	e	hassuf	1	V3SAPOXA
2	u	hassuf	1[b]	NIS2	15	ý	hassuf	1	AIS11A
3	ů	hassuf	1	NIP2	16	me	hassuf	2	V1PAPOXA
4	a	hassuf	1	NFS1	17	ou	hassuf	2	AFS41A
5	y	hassuf	1	NFS2	18	l	hassuf	1	V3SAMOMA
6	ě	hassuf	1	O1A	19	ní	hassuf	2	NNS4
7	á	hassuf	1	AFS11A	20	i	hassuf	1	NFS6
8	ho	hassuf	2[c]	AIS21A	21	ím	hassuf	2	NNS7
9	m	hassuf	1	NIS7	22	la	hassuf	2	V3SAMOFA
10	h	hassuf	1	AFP21A	23	ách	addsuf	3[d]	NFP2
11	u	addsuf	1[e]	NIS1	24	me	addsuf	2[f]	V3SAPOXA
12	é	hassuf	1	AFS21A	25	ým	hassuf	2	AIS71A
13	í	hassuf	1	V3PAPOXA					

[a]character appears in the word
[b]one-letter suffix in the word
[c]two-letter suffix in the word
[d]adding a three-letter suffix results in a word
[e]adding a one-letter suffix results in a word
[f]adding a two-letter suffix results in a word

Table 3.24: A sample of Czech lexical rules

Comparing the first fifty contextual rules learned by experiments Baltimore and Washington, just three same rules were generated at the same time. The others are similar with regard to the context size or are totally different. For instance, we present the first eight out of the mentioned fifty contextual rules in Tab. 3.26. The first rules (a) change the tag AFP21A to the tag AIP21A if the tag NIP2 appears 1 or 2 positions to the right of the

1	plavat / to swim	14	(ona) plave / she swims
2	(bez) problému / without problem	15	nový (problém) / new problem
3	(bez) problémů / without problems	16	plaveme
4	dívka / a girl	17	(s) veselou (dívkou) / with merry girl
5	(bez) dívky / without a girl	18	(on) plaval / he swam
6	pěkně / nicely	19	potápění / diving
7	veselá (dívka) / marry girl	20	(na) lodi / on the boat
8	(bez) nového (problému) / without new problem	21	(s) potápěním / with diving
9	(s) problémem / with problem	22	(ona) plavala / she swam
10	(bez) veselých (díve)k / without merry girls	23	(bez) vln / without waves
11	problém / problem	24	(on) plave / he swims
12	(bez) veselé (dívky) / without merry girl	25	(s) novým (problémem) / with new problem
13	(oni) spí / they sleep		

Table 3.25: A sample of word forms satisfying the Czech lexical rule conditions

current word token. In other words, if the current word token temporarily tagged as adjective, feminine, plural, in genitive, base form, positive is followed by two word tokens one of which is temporarily tagged as noun, masculine inanimate, plural, in genitive (i.e. there is not a `gender` agreement between the noun and its attribute), then change the `gender` value of the attribute. The second rule (b) of experiment Baltimore involves up to three preceding tags (PREV1OR2OR3TAG), while the second rule of experiment Washington only two preceding tags (PREV1OR2TAG). Similarly, the third rule (c) of experiment Baltimore involves up to three following tags (NEXT1OR2OR3TAG), while the third rule of experiment Washington only the immediately following tag (NEXTTAG). The remaining rules (d-h) operate with different information. In majority of cases, the contextual rules concentrate on the `case, gender, number` agreement. Despite the relatively small training data size, the results of RB tagger (see Fig. 3.4) on Czech seem very optimistic when compared with the results (74.20%) of statistical experiment Copenhagen trained on data of a comparable size.

3.4.3 Xerox Strategy

The numbers representing the results of all Xerox experiments are presented in Fig. 3.5. Given the strategy of Xerox taggers, they belong to the set of taggers with morphological preprocessing. The tag set size is the aspect we

	Baltimore			
	Change	to	if	
a	AFP21A	AIP21A	NEXT1OR2TAG	NIP2
b	NIS2	NIS6	PREV1OR2OR3TAG	Rv
c	AIS21A	ANS21A	NEXT1OR2OR3TAG	NNS2
d	AIP21A	AFP21A	NEXT1OR2TAG	NFP2
e	AFS41A	AFS71A	NEXT1OR2OR3TAG	NFS7
f	NIS2	NFS4	PREVTAG	AFS41A
g	AFS21A	AFS61A	NEXTTAG	NFS6
h	NFS2	NFS6	PREV1OR2OR3TAG	Rv
	Washington			
	Change	to	if	
a	AFP21A	AIP21A	NEXT1OR2TAG	NIP2
b	NIS2	NIS6	PREV1OR2TAG	Rv
c	AIS21A	ANS21A	NEXTTAG	NNS2
d	NNS2	AFS21A	NEXTTAG	NFS2
e	AFP21A	AFP61A	NEXTTAG	NFP6
f	AFS21A	ANS11A	PREV1OR2OR3TAG	V3SAPOXA
g	NNS2	V3SAPOXA	PREV1OR2OR3TAG	T_IP
h	NNS2	NNS3	PREV1OR2OR3TAG	Rk

Table 3.26: A sample of Czech contextual rules

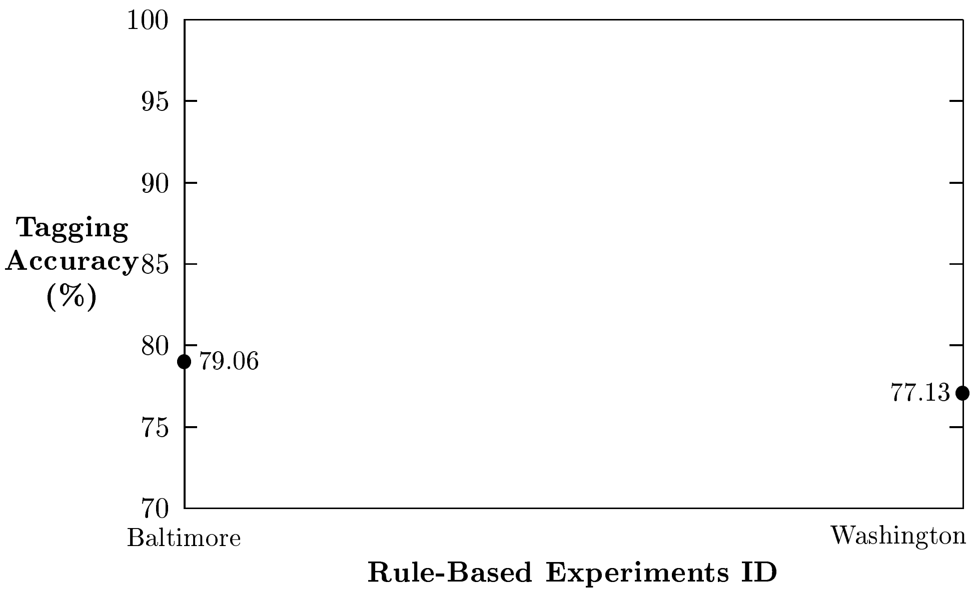

Figure 3.4: The results of the rule-based experiments

concentrate our attention on.

EXPERIMENT ID	CORPUS	TRAINING DATA SIZE
Montreal	$\mathrm{CNC}^{47}_{Xerox}$	15,000
Philadelphia	$\mathrm{CNC}^{43}_{Xerox}$	15,000
Grenoble	$\mathrm{CNC}^{34}_{Xerox}$	15,000

Table 3.27: The specification of the Xerox experiments

tag set size The results show that the more radical reduction of Czech tags (CTC tag set $\rightarrow$ Xerox_{34} tag set) the higher accuracy of the results and the more comparable are the Czech and English (Tab. 2.3) results. Again, we face the problem of the amount of information provided by the tags. For instance, how can we grammatically check the input Czech sentence when the only information we are provided (with precision 96%) is the part of speech information?

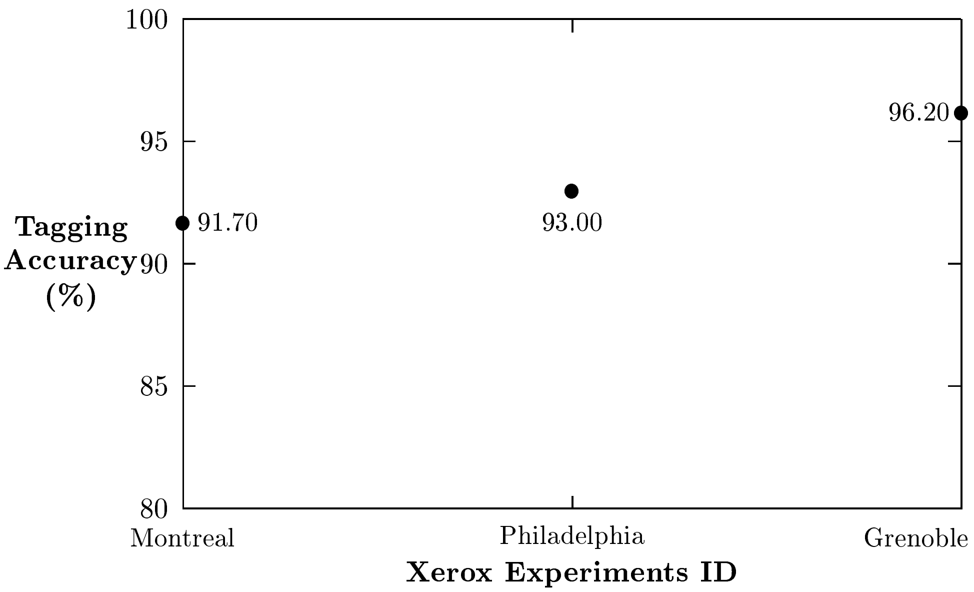

Figure 3.5: The results of the Xerox experiments

3.4.4 EXP Strategy

The strategy of an exponential tagger is driven by the set of features which can operate on any context (in general). Within the experiment Tihany and Tűbingen, we limited the pool of contexts to a combination of:

- currently processed word form, or
- AC of a single category

and

- the current position in text, or
- the immediately preceding/following position in text, or
- the closest preceding/following position (up to four positions away) in text having a certain AC in the POS category.

The use of the VTC should guarantee an improvement of overall accuracy, but the accuracy of the individual MCs is open. The experiments (Fig. 3.6) support the hypothesis on the overall accuracy.

EXPERIMENT ID	STRATEGY	CORPUS	TRAINING DATA SIZE
Tihany	pre-determined context only	PDT 0.5	198,023
Tűbingen	VTC	PDT 0.5	198,023

Table 3.28: The specification of the exponential experiments

3.5 Discussion of the Results

The series of the experiments came into life step by step and each experiment was intended to improve to improve the tagging accuracy of the previous experiments. The increasing character of the accuracy curves shows that we have been successful in the selection of the model 'parameters' - *more training data, a less detailed tag set, a different tagging method, inclusion of linguistic information.* The choice of such 'parameters' has emerged mostly from the comparison of our different approaches to tagging. The experience from other tagging experiments had a very important influence on our decisions as well. The results show that a smaller tag set achieves better tagging

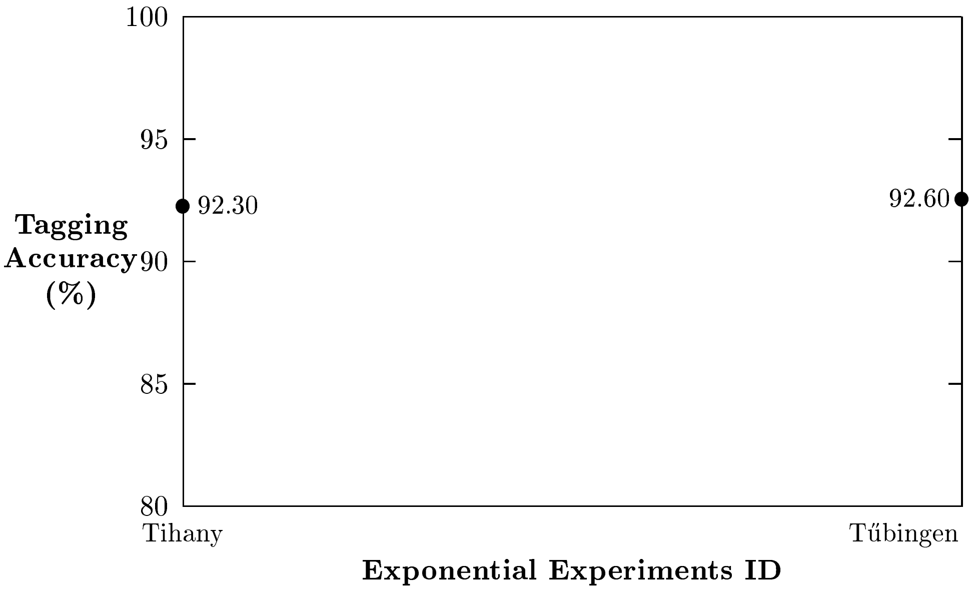

Figure 3.6: The results of the exponential experiments

performance than the bigger one does (as expected) and the statistical approach seems to be a little better than a rule-based one. Nevertheless, the results mean that many sentences will contain at least one error.

None of the representative corpus-based tagging methods do achieve the magic point of 100% performance. It is believed that the context can reveal almost all the secrets of a language. We stress *almost*, in some cases the context is not sufficient to specify the function/meaning of a word form. As we are interested in context-based models of language, the magic point of such models cannot be 100% because the world knowledge which is hidden somewhere between the lines cannot be read from the set of word forms and tags.

Using Xerox tagging tools, the tagging accuracy (Grenoble - 96.2%) is becoming closer to 98%. However, the Xerox experiment was performed upon a smaller tag set containing tags concentrating mostly on POS classes and, not in all but in many applications, it is too coarse for the subsequent processing of the tagged text such as automatic syntactic analysis, spelling correction, speech recognition, etc.

One of the conclusions, which we have drawn from the experiments, is the following: the tag set should be chosen according to the requirements of a given application rather than to optimize it for the tagger. The more detailed tag set the better - but again, one should primarily consider the application at hand and (if possible at all) must optimize the accuracy/tag set size ratio.

All results reported in Table 3.29[5] are based on the best-only approach using the tagging accuracy criterion. It should be stressed that whereas the experiments Hlinsko, Prague, Mariánská, Copenhagen, Granada and London do not use any morphological preprocessing, the experiments Birmingham, Montecatini, Montreal, Philadephia, Grenoble, Tihany, Tűbingen, Mannheim, EXP_CZ and MM_CZ$_{bi}$ employ the output of MA; the experiments Baltimore, Washington learn the morphology from a part of the training data. The experiments Mariánská and Mannheim illustrate the importance of morphological preprocessing; in spite of the double size of training data in the experiment Mariánská with regard to the Mannheim training data size, the TA of Mannheim experiment significantly (80.91% vs. 93.38%) exceeds the TA of Mariánská experiment. The main conclusion is that the best tagging results (93.85%) were achieved using the exponential tagger.

[5] Table 3.29 recapitulates not only the experiments discussed above, but the experiments performed later on the PDT as well. The experiments with the best TA within the characteristic subset of experiments are in boldface.

STRATEGY	EXPERIMENT ID	CORPUS	TRAINING DATA SIZE	TAGGING ACCURACY (%)
unigram MM	Hlinsko	CTC	621,015	65.53
bigram MM	**Prague**	**CTC**	**621,015**	**81.38**
trigram MM	Mariánská	CTC	612,015	80.91
bigram MM	Copenhagen	CTC	110,874	74.20
bigram MM	Granada	RCTC	621,015	90.11
trigram MM	**London**	**RCTC**	**612,015**	**90.30**
bigram MM + MA	**Birmingham**	**PDT 0.5**	**124,692**	**87.80**
trigram MM + MA	Montecatini	PDT 0.5	124,692	86.90
bigram MM + MA	$\mathrm{MM\_CZ}_{bi}$	PDT 0.5	300,000	91.80
trigram MM + MA	$\mathrm{MM\_CZ}_{tri}$	PDT 0.5	300,000	92.80
RB	**Baltimore**	**CTC**	**75.863**	**79.06**
RB	Washington	CTC	19.565	77.13
Xerox_{47}	Montreal	$\mathrm{CNC}^{47}_{Xerox}$	15,000	91.70
Xerox_{43}	Philadelphia	$\mathrm{CNC}^{43}_{Xerox}$	15,000	93.00
Xerox_{34}	**Grenoble**	**$\mathrm{CNC}^{34}_{Xerox}$**	**15,000**	**96.20**
pre-determined context only EXP	Tihany	PDT 0.5	198,023	92.30
VTC EXP	Tűbingen	PDT 0.5	198,023	92.60
EXP	**EXP_CZ**	**PDT 0.5**	**300,083**	**93.85**
trigram MM + MA EM alg.	**Mannheim** ($\mathrm{MM\_CZ}_{tri}$)	**PDT 0.5**	**300,083**	**93.38**
bigram MM + MA EM alg.	$\mathrm{MM\_CZ}_{bi}$	PDT 0.5	300,083	92.50

Table 3.29: The overview of Czech tagging experiments

Chapter 4

Context Considerations

Notwithstanding the importance of context information, we have not included the context considerations into the evaluation of the influence of various parameters on the tagging accuracy yet. Let us do it now. Our aim is to focus on the problem, which context should be selected from the processed text to tag it properly rather than to concentrate on the way by which the tagging strategies treat the context information. The PDT is intended to serve as the source of annotated data.

4.1 English and Czech Tagging Experiments

The corpus-based approaches determine the amount of human work involved in the NLP tasks on the building of training data and on the coming up with an algorithm giving results as precise as possible. The ideas of context specification cannot be left out in the formulation of the algorithm. The scope: of context must be specified according to the character of the particular NLP task. As we consider the nature of context from the perspective of the tagging application, an elementary unit we process is a word token. In general, there is no strict rule saying how many preceding and following word tokens we should look at to be sure that we tag the word token properly. Thus, let us have a look at the empirical experience.

Let $W = w_1 w_2 w_3 ... w_n$ be an input text to be tagged. As all the presented tagging strategies tag the input text in the left-to-right direction, a word token w_i is processed when the word tokens $w_1 w_2 ... w_{i-1}$ have already been tagged - $w_1|t_1 ... w_{i-1}|t_{i-1} w_i w_{i+1} ... w_n$[1]. For the currently processed word token w_i, the context $c(w_i)$ of the representative corpus-based tagging strategies for tagging Czech (MM_CZ$_{tri}$, MM_CZ$_{bi}$, EXP_CZ taggers - see Tab. 3.29) and English (MM_EN, ME_EN, EXP_EN, MB_EN, RB_EN, NE_EN taggers[2]) can be expressed as follows:

[1] As the EXP tagger operates on a subtag level determined by the ACs, (see Sec. 2.3) let ma$_i$ consist of all fifteen ACs for a given word token w_i. The text entering the tagging step has the form $\{w_1, ma_1, w_2, ma_2, ..., , w_n, ma_n\}$

[2] The basic characteristics of the given corpus-based English tagging experiments are

MM_EN - $c(w_i) = \{w_{i-2}, t_{i-2}, w_{i-1}, t_{i-1}\}$

MM_CZ$_{tri}$ - $c(w_i) = \{t_{i-2}, t_{i-1}\}$

MM_CZ$_{bi}$ - $c(w_i) = \{t_{i-1}\}$

ME_EN - $c(w_i) = \{w_{i-2}, t_{i-2}, w_{i-1}, t_{i-1}, w_{i+1}, w_{i+2}\}$

EXP_CZ, EXP_EN - $c(w_i) = \{w_{i-4}, ma_{i-4}, w_{i-3}, ma_{i-3}, w_{i-2}, ma_{i-2}, w_{i-1}, ma_{i-1}, w_{i+1}, ma_{i+1}, w_{i+2}, ma_{i+2}, w_{i+3}, ma_{i+3}, w_{i+4}, ma_{i+4}\}$

MB_EN - not directly specified

RB_EN - $c(w_i) = \{w_{i-3}, t_{i-3}, w_{i-2}, t_{i-2}, w_{i-1}, t_{i-1}, w_{i+1}, w_{i+2}, w_{i+3}\}$

NE_EN - $c(w_i) = \{w_{i-3}, t_{i-3}, w_{i-2}, t_{i-2}, w_{i-1}, t_{i-1}, w_{i+1}, w_{i+2}\}$

Observing the given description, the Markov models are locally (processing w_i) based only upon the left-hand side context[3] while the other strategies look not only at the left positions but consider also the right-hand side context. Some authors offer practical experience with a modification of the context scope. In the paper [Schmid, 1994], the author describes the context shrinking to two preceding and one following words together with their tags which causes accuracy reduction only by 0.1%. Enlarging the context gave no improvement. The authors [Daelemans, Zavrel, 1996] do not specify directly the context scope, but they construct a distance metrics between similar enviroments within modest contexts. We can conclude that the enumerated contexts as a whole are limited up to 4 positions to the left/right.

4.2 The Context for Humans

At the starting point of the tagging procedure, all tagging strategies are given the same input text. The input text (as a whole) is understood as a whole text context. Consequently, the tagging strategies select from the whole text context any subcontext over which they process the given word token. Let us limit the subcontext to the word tokens $(w_1, w_2, ..., w_{i-1})$ preceding the currently processed word token (w_i) within the input text. For a vocabulary size n there are n^{i-1} different subcontexts (for ex. $n = 1{,}000$ and $i = 4$ then $n^3 = 10^9$). The problem which immediately appears concerns

introduced in Tab. 2.3.

[3] In the end, the incorporation of the Viterbi algorithm to find the best tag sequence means the usage of the right-hand side context.

the matrices (of n^{i-1} order) representing the counts of particular subcontexts within the training corpus. With regard to the astronomically large number of such subcontexts, a vast majority of the possible subcontexts will never occur in Czech (or other natural language) and that is the reason why the given matrices are sparse. Nevertheless, the computational linguists' effort is directed to deal with sparseness of data being connected with context specification.

The bigram and trigram MMs employ the smallest left-hand side context size relatively to the other corpus-based methods; at the same time, their performances are the best (Tab. 2.3, Tab. 3.29). We believe that a further improvement of MMs lies in a better selection of the analysed context. Not to limit ourselves only to experiments modifying the context size and in order to discover certain guidelines we explore how people handle the information coming from the predefined left-hand side context.

4.2.1 Prerequisites

The annotation of the test file was assigned to a group of 5 students: 2 undergraduate students (S1, S2) with rich experience learned during the annotation of the PDT; 3 computational linguistics graduate students (S3, S4, S5) - one of them (S5) with an experience with various tagging strategies and one of them (S4) being bilingual not educated in Czech. The test file that was given to the students comprised a 283 word token subset (141 unambiguous tokens and 142 ambiguous tokens) of the test file which we used in the tagging experiments mentioned above (MM_CZ$_{tri}$, MM_CZ$_{bi}$, EXP_CZ). For purposes of evaluation of the tagging and annotation, the given test file was annotated independently by another annotator upon an unlimited context.

Formalism

Let S $= w_1 w_2 ... w_s$ be a sentence[4] (a sequence of word tokens) we tag/annotate in the left-to-right direction, $S_{tokens} = (w_i)_{i=1..s}$ be a list of word tokens occurring in the sentence S. While tagging the *i*-th word, the *i-1* preceding word tokens are already tagged by tags $t_1, t_2, ..., t_{i-1}$; let T be a list of tags $(t_j)_{j=1...i-1}$.

The contexts which come into play during the experiments of annotation (BC, TC, SC) and the experiments of tagging (TTC, BTC) can be defined as functions:

[4] We consider a context within a sentence, we do not cross the sentence boundaries.

- **Bigram Context (BC)** as a function

  $BC : S_{tokens} \rightarrow 2^{S_{tokens}}, BC(w_i) = \{w_{i-1}\}, BC(w_1) = \emptyset$

- **Tag Bigram Context (TBC)** as a function

  $TBC : S_{tokens} \rightarrow 2^{T}, TBC(w_i) = \{t_{i-1}\}, TBC(w_1) = \emptyset$

- **Trigram Context (TC)** as a function

  $TC : S_{tokens} \rightarrow 2^{S_{tokens}}, TC(w_i) = \{w_{i-1}, w_{i-2}\}, TC(w_1) = \emptyset,$
  $TC(w_2) = \{w_1\}$

- **Tag Trigram Context (TTC)** as a function

  $TTC : S_{tokens} \rightarrow 2^{T}, TTC(w_i) = \{t_{i-1}, t_{i-2}\}, TTC(w_1) = \emptyset,$
  $TTC(w_2) = \{t_1\}$

- **Sentence Context (SC)** as a function

  $SC : S_{tokens} \rightarrow 2^{S_{tokens}}, SC(w_i) = \{w_1, ..., w_{i-1}\}, SC(w_1) = \emptyset$

To illustrate the defined terms, let us assume a sample of the sentence fragment *O další Stříbrné medvědy se podělily ...* [lit. about – further – Silver – Bears – Refl. – they-shared ..., E. The remaining (Prizes of) Silver Bears were obtained by ...] and let us suppose that the first four word tokens are already tagged O|RR--4---------- další|AAMP4----1A---- Stříbrné|AAMP4----1A---- medvědy|NNMP4-----A----. Then the word token *se* is to be tagged/annotated. According to the chosen particular context, the word token *se* is being processed within the context information embodied in one of the sets BC(se) = {medvědy}, TC(se) = {Stříbrné, medvědy}, SC(se) = {O, další, Stříbrné, medvědy}, TTC(se) = {AAMP4----1A----, NNMP4-----A----}, BTC(se) = {NNMP4-----A----}.

4.2.2 How Humans Treat the Context Information

A specially developed tool for morphological annotation, which offers an easy disambiguation of lemmas and tags (which are outputs of the automatic morphological analysis), was used as a disambiguation tool, which displays, for the currently annotated ambiguous word token, its morphological information and the whole text context. For our aims, we have modified the disambiguation tool in the sense of the visibility of a partial context; in case of Bigram Context only the previous word token is visible, for Trigram Context only two previous word tokens are, and finally, for Sentence Context the preceding word tokens up to the beginning of the sentence are at the annotator's disposal. We have to stress that unambiguous word tokens remain obviously untouched by the annotator and while annotating the given ambiguous word token the annotators have no information on the assigned tags of the word tokens which are included in the context; annotators just suggest a hypothesis of the tags of the context word tokens themselves. On the other hand, the presented Markov models working over Tag Trigram/Bigram Context do not deal with the word tokens.

4.3 Discussion of the Results

Table 4.1 provides information on the evaluation of the annotation and tagging of the given test file. Reading the table horizontally, we observe that all the students are getting better as the context enlarges. Reading the table vertically, we speculate that the learned experience in the course of the annotation over the whole context comes into play (students S1, S2 vs. students S3, S4, S5). On the other hand, the knowledge of the tagging methods seems not to be so important (student S5). The bigram MMs beat the students annotating over the bigram context TBC. However, the situation is inverse for the trigram contexts TTC, TC - annotation almost beats tagging.

context	BC	TC	SC	TTC	TBC
annotator/tagger	# of incorrectly tagged/annotated ambiguous word tokens out of 142 ambiguous				
S3	36	20	16	-	-
S4	47	32	27	-	-
S1	26	20	9	-	-
S2	16	13	7	-	-
S5	29	20	17	-	-
MM_CZ$_{tri}$	-	-	-	20	-
MM_CZ$_{bi}$	-	-	-	-	24

Table 4.1: The evaluation of tagging and annotation over the predefined contexts

Table 4.2 gives a detailed view on the annotation/tagging on a subtag level[5]. A more interesting observation concerns the way how the error rate over these MCs changes as the context enlarges.

[5] We present only the most problematic MCs - `gender`, `number`, `case` - together with `POS` and `SubPOS`.

annotator/ tagger	context	POS	SubPOS	g	n	c
S3	BC	0.71	0.71	4.95	3.18	8.13
	TC	0.35	0.35	4.59	1.77	2.83
	SC	0.00	0.35	3.89	1.41	2.47
S4	BC	1.06	1.41	6.36	3.53	13.43
	TC	1.77	2.12	4.24	3.18	8.83
	SC	0.35	0.71	4.95	2.47	6.36
S1	BC	0.35	0.71	4.59	1.77	5.65
	TC	0.35	0.71	2.83	1.77	4.24
	SC	0.00	0.35	2.12	1.06	1.41
S2	BC	0.35	0.35	2.83	0.35	3.53
	TC	0.35	0.35	1.06	0.35	3.53
	SC	0.35	0.35	1.41	0.35	1.06
S5	BC	0.06	1.77	6.36	2.47	6.01
	TC	0.00	0.35	4.95	2.12	3.89
	SC	0.35	0.71	4.59	2.47	3.89
MM_CZ$_{bi}$	BTC	0.71	0.71	2.47	0.71	6.71
MM_CZ$_{tri}$	TTC	0.71	0.71	2.12	0.35	5.30

Table 4.2: Error rates (%) over the `POS`, `SubPOS`, `gender`, `number`, `case`

morphological category		g	n	c
annotator/tagger	context enlarging	the error rate improvement (%)		
	TC←BC	0.36	1.41	5.3
S3	SC←TC	0.7	0.36	0.36
	TC←BC	-0.71	0.35	4.6
S4	SC←TC	1.41	0.71	2.47
	TC←BC	1.76	0.00	1.41
S1	SC←TC	0.71	0.71	2.83
	TC←BC	1.77	0.00	0.00
S2	SC←TC	-0.35	0.00	2.47
	TC←BC	1.68	0.35	2.12
S5	SC←TC	0.36	-0.35	0.00
MM_CZ	TTC←TBC	0.35	0.36	1.41

Table 4.3: The error rate changes (%) due to the context enlarging

Looking at Tab. 4.3, the numbers represent decreasing/increasing (positive/negative numbers) of the error rates over the MCs `gender, number, case` for each student and the MM taggers. For example, for student S3, the error rate over `gender` decreases by 0.36% if the bigram context (BC) is enlarged to the trigram context (TC) and at the same time it decreases by 0.7% if the trigram context (TC) is enlarged to the sentence context (SC). Given the Czech typical word order and given the assumed left-hand side contexts, the improvement of the `case` error rate is more expressive than the changes of the `gender` and `number` error rates. Again, given the Czech word order, it is necessary to include the right-hand side context to identify the `gender` and `number` of the word token.

The strategy of human annotation described above can be understood only as a simulation of the MMs. The humans work with the left-hand side context from the beginning till the end; the MMs assign to the currently processed word token tags with regard to the left-hand side context as well, but the incorporation of the Viterbi algorithm to find the best tag sequence which means, in fact, the usage of the right-hand side context in fact.

Putting together this fact and the insufficiently representative size of the test sample we cannot make any definite conclusions. On the other hand, the presented results offer the idea that the sentence context (SC) can be sufficient for successful context-based approaches. We speculate that it is not necessary to take the sentence context (SC) as a whole, but dynamically to select a trigram subcontext from the sentence context.

Chapter 5

Classifier Combinations

The research based on the different strategies applied to tag Czech texts has reached the state in which the performance results of the Czech tagging systems are very close to each other. A relatively new idea that has emerged concerns a way of combination of Czech taggers.

In the present chapter, we first give a machine learning motivation and then, we illustrate the usage of combination techniques on concrete applications.

5.1 Motivation

Let X be a set of components $x_1, x_2, \dots, x_m$ and Y be a discrete set; we do not specify the type of the X-, Y- components. To each element x_i, we assign just one element from the set Y. In supervised learning, a set of pairs (x_i, y_i) is called a **training set**. The learning algorithm "trains" the pairs from the training set and for new input set X^{new} (be $|X^{new}| = n$) returns a set of pairs $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), x_i \in X^{new}, y_i \in$ Y. The given set of pairs is called a **classifier** and a set of classifiers is called an **ensemble of classifiers**. According to the number of different learning algorithms n, we get a set of classifiers $C = \{C_1, C_2, \dots, C_n\}$. In general, a classifier C_k (k = 1 ... n) operates on a set $X = \{x_1, x_2, \dots, x_m\}$ and is uniquely determined as the set of pairs (x_i, y_i), $x_i \in X$, $y_i \in Y$. In particular, a classifier $C_k = \{(x_1, y_1^k), (x_2, y_2^k), \dots, (x_n, y_n^k)\}$, $x_i \in X^{new}$, $y_i^k \in Y$ over the input set X^{new}.

For instance, we want to combine classifiers in the ensemble (n classifiers) to get another classifier. We apply plurality voting as the method of combination - each case x is assigned a value y for which most of the input classifiers vote. Let us suppose that *classifiers are independent, the error rates of them are equal to* p_{lim} and $p_{lim} <$ *0.5*. It is difficult to decide whether the classifiers are really independent or not and we will discuss this issue in detail below. The following question arises immediately: Does it make sense to combine classifiers in order to get a more accurate final classifier? Or, in other words, what is the probability that the value y for which most of the input classifiers vote is not the correct one? Let $\mathcal{A}$ be the event that the value y for which the most of the input classifiers vote is not the correct one, let $\mathcal{B}$ be the event that the value of y for which more than half of the input classifiers vote is not the correct one. Since the event $\mathcal{B}$ corresponds to the binomial distribution (correct/uncorrect classifier output), then

$$\Pr(\mathcal{B}) = \sum_{\lceil i=n/2 \rceil}^{n} \left(\begin{array}{c} n \\ i \end{array} \right) * p_{lim}^{i} * (1 - p_{lim})^{n-i} \tag{5.1}$$

One of the preconditions limits p_{lim} to be less than 0.5. The reason WHY is captured by Eq. 5.1. If p_{lim} equals 0.5, the accuracy of the combination is the same as the accuracy of the input classifiers; if p_{lim} is greater than 0.5, there is no way how to get a more accurate classifier by a combination of classifiers.

As the left-hand side of the implication $\mathcal{A} \subseteq \mathcal{B} \rightarrow \Pr(\mathcal{A}) \leq \Pr(\mathcal{B})$[1] comes true for our situation, the right-hand side must come true as well. Thus, let $n = 21$ and $p = 0.1$, then the error rate of the final classifier is less than or equal to 1.35e-06 which is significantly less than $p = 0.1$. [Dietterich, 1997] provides a very nice overview of the classifier combinations.

Basically, there are two types of classifiers: (i) classifiers based on different learning strategies and trained on the same training set - **original classifiers**; (ii) classifiers based on the same learning strategy but trained on different training sets. **Bagging** and **boosting** are two different methods that produce a diverse set of classifiers - **bagged** and **boosted classifiers**, respectively - by manipulating the training data used for the learning algorithm. The mentioned methods take advantage of the learning algorithm sensitivity on the training data.

The bagging algorithm ([Breiman, 1996]) votes the classifiers generated by different replicates of the original training set containing m training pairs. The different replicates are produced by a random sampling of m instances from the training set. We speak then about **bagged training sets**, which contain m training examples.

The boosting algorithm ([Freund, Schapire, 1999]) votes the classifiers generated (with regard to the distribution of correctly and incorrectly predicted training examples by the previous classifier) one after another. The classifiers work with the example weights so that the incorrectly predicted examples obtain higher weights (to put attention on the errors) than the correctly predicted examples.

[1]This is a true implication due to the definition of the probability measure.

Most of the previous writings on machine learning have demonstrated that these two methods are effective for neural networks ([Maclin, Opitz, 1997]) algorithms and decision trees ([Bauer, Kohavi, 1998], [Dietterich, 1998] and [Quinlan, 1996]). Since to find a global optimal decision tree or a neural network is NP-hard problem ([Hyafil, Rivest, 1976], [Blum, Rivest, 1988]), the given techniques use the greedy search method to find a locally optimal decision tree or locally optimal weights of a neural network. The results show that the bagging and boosting strategies compensate for the imperfection of the greedy search algorithm in the sense of improving the performance of the classifier ensembles in comparison with the performance of the individual classifiers.

In the following sections, we concentrate only on the bagging algorithm.

5.2 NLP Applications

5.2.1 Original Classifiers

Original taggers

Practically, almost every natural language processing system needs as its input a pre-tagged text. A number of various techniques has been developed to tag texts. Summarizing the results of tagging systems, the performances of different tagging approaches are comparable in the end. The character of the training data and of the strategy are the two most important things which come into play. The experiments confirm that the more training data, the higher is the tagging accuracy. An even more important aspect is the fact that the more different language patterns training data cover, the higher is the tagging accuracy. There are two possible resources to obtain better results: to have more representative training data and to apply a more “powerful” tagging method.

Similar ideas trying to improve tagging methods and using an original taggers combination are described in [Brill, 1998], [Halteren et al., 1998]. It is a well-known fact that taggers (trained on the same training data) handle the same information though in a different way with comparable results. Both papers show that all used taggers (MM, rule-based, memory-based, maximum entropy) produce different types of errors. Since the maximum entropy approach gives the best results, a maximum entropy tagger is taken as a baseline tagger. Various methods of combinations are presented - majority voting, contextual cues, memory-based techniques, etc. In [Brill, 1998], the authors achieve a maximum improvement of 0.4% over the baseline tagger using the context-based techniques. The maximum improvement of 0.48%

over the baseline tagger is increased in [Halteren et al., 1998].

In the paper [Chanod, Tapanainen, 1994], they evaluate the performances of statistical and constraint-based taggers of French not only separately but they compare their errors as well. They propose a procedure of combination of their taggers on the basis of the error evaluation, but the performance of the combination does not outperform the constraint-based tagger accuracy.

Original parsers

In [Henderson, Brill, 1999], the authors explore statistical parser[2] combination techniques. The employed techniques are divided into two basic groups; parse hybridization and parse switching. Within each group, they distinguish non-parametric (constituent voting, similarity switching) and parametric (probabilistic) strategies. The improvement of the baseline parser's performance is significant (precision up to 3%, recall up to 1%, F-measure up to 1.5%).

5.2.2 Bagged Classifiers

Bagged parsers

We present here the results of the experiments with parser bagging on Czech language using dependency structures and on English language based on phrase structures (for details, see [Hajič et al., 1998]). In the sequel we refer to the adapted Collins parser trained on the original Czech training data (Prague Dependency Treebank) as to the **Czech original parser** and to the Collins parser trained on the original English training data (Wall Street Journal) as to the **English original parser**.

In general, we carry out the following steps: first, we generate the bagged Czech/English parsers and then test the independence of the bagged parsers. The independence test of parsers A and B is measured by the percentage of dependencies/constituents posited by the parser A and not by the parser B. We take the union of all dependencies/constituents (Czech/English) at the output of bagged parsers in parsing the test data. There are two methods how to modify this union into the final **unbalanced** and **balanced** outputs. Thus, we speak about an **unbalanced** and a **balanced** method, respectively. The unbalanced output is created by keeping all dependencies/constituents that were posited by more than a half of the bagged parsers. We cannot be sure that the unbalanced output represents a fully-

[2] They used three statistical parsers.

linked parse. Thus, for every word in the sentence, the bagged parsers vote on which $X \rightarrow Y$ dependency/constituent should be chosen.

Table 5.1 recapitulates the results. For Czech, the balanced method still gives an improvement over the Czech original parser, but not as great an improvement as the unbalanced method has achieved. With 18 Czech bagged parsers we get the improvement in F-measure of 1.2% (with 6 Czech bagged parsers - 0.7%) using the unbalanced method. For English, the unbalanced method with 6 English bagged parsers gives an improvement over the English original parsers as well, but the improvement is not so "considerable" as for Czech (with 6 English bagged parsers - 0.4%). For Czech bagged parsers, the average parser independence is 7.1% and for English it is 2.2%.

Parser(s)	Precision	Recall	F-Measure
Czech original	77.0	77.0	77.0
18 Czech bagged (Balanced)	77.8	77.8	77.8
6 Czech bagged (UnBalanced)	81.1	74.6	77.7
18 Czech bagged (UnBalanced)	80.6	76.0	78.2
English original	88.7	88.4	88.5
6 English bagged (UnBalanced)	90.6	87.3	88.9

Table 5.1: Bagging results on Czech and English parsing

5.3 Tagger Independence Measure

To see how the errors produced by two taggers differ, we use a difference measure called the **complementary rate (CR)** (see [Brill, 1998]) which gives *the percentage of errors produced by a tagger A and not produced by a tagger B*:

$$CR(A,B) = (1 - Common\_Errors\_A\_B / Errors\_Produced\_by\_A) * 100(\%) \quad (5.2)$$

At the same time

$$CR(A,B) = CR(B,A) \Leftrightarrow Errors\_Produced\_by\_A = Errors\_Produced\_by\_B \quad (5.3)$$

To illustrate the defined measure, let us go back to the example given in Section 2.2. Now, as we know that $Common\_Errors\_A\_B = 1$ (w_4). Thus, CR(A,B) = (1 - 1/2)∗100(%) = 50%, CR(B,A) = (1 - 1/3)∗100(%) = 66%.

The following situations may obtain:

- **CR(A,B) = 100%** - taggers A and B are independent; combination of A and B gives the improvement.
- **CR(A,B) = 0%** - taggers A and B are totally dependent; combination of A and B does not make progress.
- **0% < CR(A,B) < 100%** - taggers A and B are dependent; the higher CR(A,B) the stronger is the chance to get better results by the combination of A and B.

Chapter 6

Bagging Czech Taggers

Early studies point out that if bagging is to succeed, it must produce a sufficiently diverse set of classifiers and the ensemble of classifiers should not degrade the original classifier performance significantly. A diverse set of classifiers can be explained as a set of classifiers producing different types of errors, better to say complementary errors.

Whether bagging can be a good solution of the Czech tagging problem remains an open question. The answer to such a question should be provided by the set of the experiments we perform.

6.1 Data

The manually disambiguated text file we use for experiments with bagging Czech taggers comprises 23,397 sentences and 330,355 words excluding punctuation marks. We divide the file into two parts. The first part, called the **original training set**, consists of 21,270 sentences and 300,383 words (excluding punctuation marks), compiled by taking the first ten of every eleven sentences. This part is used to generate **bagged training sets** of the original training set according to the bagging strategy. The second part, the **test data**, consists of 2,127 sentences and 29,972 words, created by taking the last one of every eleven sentences.

In the first bunch of experiments, we have generated totally **20** bagged training sets (train_1.dat, train_2.dat, ... , train_20.dat) from the original training set (train_0.dat). Table 6.1 provides an overview of their basic characteristics. Given the bagging strategy and given the precondition that the sentence is taken as a training example, the number of sentences in all bagged training sets is the same (i.e. 21,270). The bagged training sets cover on average 300,000 words (again, excluding punctuation marks). The last column of Table 6.1 illustrates the fact that each of the bagged training sets contains about 73% unique sentences from the original training set, i.e. the remaining 27% sentences appear in the bagged training sets at least twice.

	# OF WORDS (EXCLUDING PUNCTUATION)	# OF SENTENCES	# OF UNIQUE SENTENCES
ORIGINAL TRAINING SET	300,383	21,270	21,270
BAGGED TRAINING SETS			
1	300,019	21,270	15,640
2	301,484	21,270	15,624
3	301,063	21,270	15,578
4	301,989	21,270	15,674
5	299,389	21,270	15,648
6	300,450	21,270	15,639
7	303,207	21,270	15,647
8	302,774	21,270	15,719
9	299,842	21,270	15,636
10	299,170	21,270	15,645
11	299,679	21,270	15,578
12	301,368	21,270	15,587
13	299,711	21,270	15,640
14	300,913	21,270	15,641
15	299,371	21,270	15,704
16	302,509	21,270	15,655
17	298,846	21,270	15,702
18	298,680	21,270	15,613
19	300,609	21,270	15,630
20	302,608	21,270	15,624

Table 6.1: Original and bagged training set characteristics

6.2 Taggers

We use the tagger based on the Markov model strategy ([Mírovský, 1998], Sect. 3.4.1). In the sequel, we will refer to the MM tagger trained on the original training set (train_0.dat) as the original MM tagger (or tagger $T0_{MM}$) and to the MM tagger trained on the bagged version of the training set as the bagged MM tagger (or tagger $T[1\text{-}9][0\text{-}9]_{MM}$); for instance, the Markov model tagger $T5_{MM}$ is trained on the bagged training set train_5.dat.

6.3 MM Taggers Error Analysis

6.3.1 Tag Level Errors

Table 6.2 displays not only the tagging accuracy of all (21) MM taggers we work with, but also the total number of errors on the tag and the subtag levels. Reading across the first line, the table shows that the tagging accuracy of the original MM tagger is 91.53% ($\equiv$ 2,540 words from 29,972 were tagged incorrectly); at the same time each incorrectly assigned tag averages out at almost two mistagged *MC*s. In the case of the original MM tagger it means 4,872 mistagged *MC*s.

The original MM tagger is the best one when comparing tagger accuracies on the tag level and the accuracies of the bagged taggers range from 90.75% to 91.26% . It is an optimistic observation that the difference in performance between the original tagger and any single bagged tagger is not extremely high. Also, the difference in tagging accuracy among the 20 bagged taggers is really very small ($\leq$0.5).

As has been said above, a combination of classifiers can increase the system performance if classifiers entering the combination are mutually complementary (i.e. they produce different types of errors). Tables 6.3, 6.4 and 6.5 provide *CR*s of MM taggers[1]. As the original tagger has the highest tagging accuracy, the numbers in the first line are low with regard to the others. That is understandable because the highest tagging accuracy means the lowest number of errors. All complementary rates are non-zero numbers and thus we can conclude that bagged taggers are somehow complementary and that there is a reasonable chance to get a higher tagging accuracy by combining taggers.

[1]For example, CR($T0_{MM}$, $T8_{MM}$ = 12.32%), CR($T10_{MM}$, $T19_{MM}$ = 17.24%)

Tagger	Accuracy (%)	# of tag level errors	# of subtag level errors
$T0_{MM}$	91.53	2,540	4,872
$T1_{MM}$	90.75	2,771	5,259
$T2_{MM}$	90.98	2,704	5,168
$T3_{MM}$	90.81	2,753	5,204
$T4_{MM}$	90.97	2,707	5,147
$T5_{MM}$	91.17	2,647	5,096
$T6_{MM}$	91.06	2,678	5,084
$T7_{MM}$	91.01	2,693	5,180
$T8_{MM}$	90.89	2,729	5,257
$T9_{MM}$	90.83	2,747	5,199
$T10_{MM}$	91.05	2,681	5,086
$T11_{MM}$	90.96	2,709	5,237
$T12_{MM}$	90.98	2,703	5,161
$T13_{MM}$	91.09	2,670	5,049
$T14_{MM}$	90.75	2,771	5,198
$T15_{MM}$	91.08	2,673	5,115
$T16_{MM}$	91.02	2,691	5,169
$T17_{MM}$	91.01	2,694	5,194
$T18_{MM}$	90.91	2,724	5,176
$T19_{MM}$	90.94	2,715	5,221
$T20_{MM}$	91.26	2,620	5,084

Table 6.2: Accuracy of the MM taggers

	$T0_{MM}$	$T1_{MM}$	$T2_{MM}$	$T3_{MM}$	$T4_{MM}$	$T5_{MM}$	$T6_{MM}$	$T7_{MM}$	$T8_{MM}$	$T9_{MM}$
$T0_{MM}$	0.00	9.92	11.65	9.69	11.02	11.69	10.39	12.28	12.32	10.39
$T1_{MM}$	17.43	0.00	19.45	16.82	18.33	18.44	18.33	19.92	18.80	18.40
$T2_{MM}$	17.01	17.46	0.00	17.71	18.16	18.75	18.20	19.27	18.53	17.23
$T3_{MM}$	16.67	16.27	19.18	0.00	18.34	18.45	18.27	19.22	18.38	17.62
$T4_{MM}$	16.51	16.40	18.25	16.96	0.00	18.10	17.55	17.55	17.25	17.44
$T5_{MM}$	15.26	14.62	17.00	15.19	16.24	0.00	16.96	17.26	16.66	15.64
$T6_{MM}$	15.01	15.50	17.40	15.98	16.65	17.92	0.00	17.51	16.47	16.09
$T7_{MM}$	17.27	17.60	18.94	17.42	17.12	18.68	17.97	0.00	19.61	17.71
$T8_{MM}$	18.40	17.55	19.27	17.66	17.92	19.16	18.03	20.67	0.00	19.13
$T9_{MM}$	17.15	17.69	18.53	17.44	18.64	18.71	18.20	19.33	19.66	0.00
$T10_{MM}$	16.30	16.45	18.39	16.71	18.58	17.61	17.31	18.05	18.02	18.05
$T11_{MM}$	16.39	15.58	19.27	17.09	17.46	18.60	17.53	18.86	18.79	17.76
$T12_{MM}$	16.28	16.24	19.13	16.35	18.02	18.28	17.61	18.76	17.94	16.57
$T13_{MM}$	15.32	15.28	17.75	15.73	17.34	17.60	17.15	19.10	17.04	16.74
$T14_{MM}$	17.72	17.79	19.60	17.39	18.15	20.28	18.73	20.17	18.19	18.55
$T15_{MM}$	15.60	16.54	17.92	16.12	16.50	17.66	17.36	18.44	17.99	17.06
$T16_{MM}$	16.91	16.20	18.51	15.87	17.43	17.58	17.13	18.69	18.06	17.32
$T17_{MM}$	15.40	15.26	16.52	16.74	17.71	17.89	16.82	17.82	17.85	17.37
$T18_{MM}$	17.07	17.58	18.58	16.48	17.80	19.38	17.55	19.53	17.66	17.58
$T19_{MM}$	16.61	16.54	18.53	16.65	17.20	17.38	17.53	19.04	18.23	16.65
$T20_{MM}$	15.92	16.15	16.15	16.18	16.53	17.79	17.40	18.17	17.33	16.95

Table 6.3: Complementary rates (%): part I

	$T10_{MM}$	$T11_{MM}$	$T12_{MM}$	$T13_{MM}$	$T14_{MM}$	$T15_{MM}$
$T0_{MM}$	11.65	10.83	10.91	10.98	10.24	11.18
$T1_{MM}$	19.16	17.47	18.30	18.37	17.79	19.49
$T2_{MM}$	19.08	19.12	19.16	18.79	17.60	18.86
$T3_{MM}$	18.89	18.42	17.87	18.27	16.85	18.56
$T4_{MM}$	19.36	17.40	18.14	18.47	16.22	17.55
$T5_{MM}$	16.55	16.70	16.55	16.89	16.55	16.85
$T6_{MM}$	17.21	16.58	16.84	17.40	15.91	17.51
$T7_{MM}$	18.42	18.38	18.46	19.79	17.86	19.05
$T8_{MM}$	19.46	19.38	18.72	18.83	16.93	19.68
$T9_{MM}$	20.02	18.89	17.91	19.08	17.84	19.29
$T10_{MM}$	0.00	18.39	17.75	18.02	17.01	18.76
$T11_{MM}$	19.23	0.00	18.05	18.01	17.42	18.97
$T12_{MM}$	18.42	17.87	0.00	19.05	17.46	18.50
$T13_{MM}$	17.68	16.82	18.05	0.00	15.77	17.68
$T14_{MM}$	19.70	19.27	19.49	18.84	0.00	19.78
$T15_{MM}$	18.52	17.88	17.58	17.77	16.84	0.00
$T16_{MM}$	17.61	17.58	18.17	17.91	17.43	18.10
$T17_{MM}$	18.08	16.78	17.59	17.22	15.52	18.26
$T18_{MM}$	17.91	18.28	19.49	18.47	16.45	18.32
$T18_{MM}$	18.38	18.27	18.05	18.12	16.76	18.45
$T20_{MM}$	16.95	16.87	17.29	17.21	16.34	17.37

Table 6.4: Complementary rates (%): part II

	$T16_{MM}$	$T17_{MM}$	$T18_{MM}$	$T19_{MM}$	$T20_{MM}$
$T0_{MM}$	11.97	10.28	11.06	10.87	13.27
$T1_{MM}$	18.62	17.61	18.98	18.22	20.71
$T2_{MM}$	18.90	16.83	17.97	18.20	18.75
$T3_{MM}$	17.76	18.53	17.36	17.80	20.23
$T4_{MM}$	17.92	18.10	17.29	16.96	19.21
$T5_{MM}$	16.21	16.43	17.04	15.26	18.62
$T6_{MM}$	16.73	16.32	16.13	16.39	19.19
$T7_{MM}$	18.75	17.79	18.60	18.38	20.39
$T8_{MM}$	19.20	18.91	17.81	18.65	20.63
$T9_{MM}$	19.00	18.97	18.27	17.62	20.79
$T10_{MM}$	17.31	17.68	16.60	17.34	18.84
$T11_{MM}$	18.12	17.24	17.83	18.09	19.60
$T12_{MM}$	18.53	17.87	18.87	17.68	19.83
$T13_{MM}$	17.27	16.48	16.82	16.74	18.76
$T14_{MM}$	19.81	17.86	17.86	18.44	20.89
$T15_{MM}$	17.55	17.62	16.76	17.17	19.00
$T16_{MM}$	0.00	17.69	16.50	17.09	18.91
$T17_{MM}$	17.78	0.00	16.93	16.82	18.75
$T18_{MM}$	17.51	17.84	0.00	18.91	20.56
$T18_{MM}$	17.83	17.46	18.64	0.00	20.11
$T20_{MM}$	16.72	16.45	17.40	17.21	0.00

Table 6.5: Complementary rates (%): part III

6.3.2 Subtag Level Errors

Tagging test data by the original tagger and by the 20 bagged taggers, we obtain 20+1 tagged versions of the test data. At the point of error analysis, we are interested just in the incorrectly tagged words. We convert the set of all incorrectly assigned tags into 6-position strings - **ABCDEF** - in order to have for each tagged version of test data (20+1) a more comfortable data format and thus to be able to analyse varied kinds of errors. A string ABCDEF contains the following values:

A	tagger_output_`POS`	# tagged part of speech of the word
B	truth_`POS`	# annotated part of speech of the word
C	position_of_mistagged_*MC*	# 1...15
D	tagger_output_*MC*	# tagged value of *MC* in the given position
E	truth_*MC*	# annotated value of *MC* in the given position
F	word_order_in_testdata	# word position in the test data

Using the regular expression syntax, we can formulate various kinds of queries to obtain an error-related statistics. For instance, if we are interested in `case` (5th position) errors of words tagged as a *noun* (N) we can express it as `N.5..[0-9]*`; if we are interested in all cases where *masculine animate* (M) *adjective* (A) is tagged as *masculine inanimate* (I) *adjective* (A), then we search for `AA3IM[0-9]*`.

Let us formulate a regular expression (using the defined format) "matching" the errors over subtag `POS` (i.e. incorrect `POS` subtag in the first position). The corresponding regular expression is `.[ACDIJNPRTVX]1..[0-9]*`. Table 6.6 displays the evaluation of this expression for original and bagged taggers. Reading across the third line, Table 6.6 shows that the bagged tagger $T2_{MM}$ mistagged adjectives 46 times, numerals 47 times, adverbs 20 times, interjection once, mistagged conjunctions 12 times, nouns 70 times, etc. Totally, the bagged tagger $T2_{MM}$ mistagged MC `part of speech` of 362 words (last column).

Typically, we calculate the tagging accuracy of taggers on the tag level. On the subtag level, we are interested in the error rates on each particular *MC*. The columns in Tables 6.9 and 6.10 are marked by *MC* variables (totally, 13 *MC*s, see above) and each table box displays the error rate of the original/bagged tagger on the given *MC*. For instance, the error rate of the tagger $T20_{MM}$ on `gender` is 2.52%. The column marked by `POS` contains not only the error rate on the `part of speech` category but the number of words with incorrectly tagged subtag `POS` as well. Obviously, these numbers are the same as the numbers in Table 6.6 in the column marked by $\sum$. Even now it is certain that *MC*s `case, number` and `gender` are the most problematic categories; the error rates on them are by far the highest ones.

Again, in terms of the 6-position string format, the regular expression `AA3..[0-9]`$*$ matches strings describing the situation when an adjective is tagged as an adjective but `case` (of adjective) is mistagged. Using regular expressions like this (i.e. `part of speech` of word is correct, but some of *MC*s are incorrect), we show the results of their evaluation in Tables B.1 - B.10 (Appendix B)[2] which confirm the already known fact that morphological categories `case, number` and `gender` belong to the most problematic Czech morphological categories (from the point of view of morphology). Their high error rates bring about the question whether their complementary rates are high as well. Tables C.1 - C.9 (Appendix C) give the answers.

Tables C.1 - C.3 display `case` CRs which are mostly greater than 70%; this sounds encouragingly high. `number` CRs in the Tables C.4 - C.6 are overall smaller in comparison with `case` and `gender` CRs. However, `number` CRs average out at more than 50% which is from the point of view of tagger combination a reasonably high number as well. Like `case` CRs, `gender` CRs in Tables C.7 - C.9 are very close to the limit of (on average) 70%.

The percentages in the previous tables look very optimistic. In this connection, we are interested in mutual erroneousness of *MC*s. The information[3] in Table 6.11 must be read so that original MM tagger assigned wrong `POS,SubPOS,g,n,c,a` values (all at once) to 99 words or wrong `n,c` values to 332 words. The original MM tagger mistagged 2,540 words. Using data such as these, we can again confirm the fact that the `case, number` and `gender` belong to the most often mistagged *MC*s. Also, it is necessary to remark that `case`, `number` and `gender` errors often "go" together.

6.4 Combination

It is customary to convert the output of taggers (not only the errors, see above) into a more "comfortable" format. The formalism chosen is explained below.

[2] We present only those MCs which are included in the inflection (in the course of Czech AMA) of the given part of speech.

[3] We list those situations that occur more than 10 times.

Tagger	A	C	D	I	J	N	P	R	T	V	X	$\sum$
$T0_{MM}$	44	42	25	1	12	68	86	18	40	15	11	362
$T1_{MM}$	48	46	25	1	13	71	91	20	39	17	12	383
$T2_{MM}$	46	47	20	1	12	70	90	24	38	17	11	376
$T3_{MM}$	44	47	23	1	13	68	90	20	38	17	10	371
$T4_{MM}$	43	43	25	1	12	68	91	20	38	20	9	370
$T5_{MM}$	50	42	24	1	13	72	86	23	39	16	11	377
$T6_{MM}$	44	44	24	1	13	68	87	15	42	17	10	365
$T7_{MM}$	49	47	25	1	12	70	88	22	37	18	10	379
$T8_{MM}$	52	53	27	1	12	70	87	20	38	17	8	385
$T9_{MM}$	49	46	21	1	14	71	85	17	42	15	9	370
$T10_{MM}$	45	44	25	1	12	71	88	22	42	15	9	374
$T11_{MM}$	49	45	22	1	12	75	92	21	44	20	10	391
$T12_{MM}$	47	43	25	1	13	70	90	27	38	16	11	381
$T13_{MM}$	43	45	21	1	14	66	90	17	40	15	10	362
$T14_{MM}$	47	46	23	1	12	67	94	21	43	16	8	378
$T15_{MM}$	46	46	23	1	13	69	92	22	36	20	8	376
$T16_{MM}$	48	49	23	1	12	71	94	23	35	18	8	382
$T17_{MM}$	50	48	23	1	13	69	93	21	41	17	8	384
$T18_{MM}$	54	49	21	1	13	70	93	19	38	15	9	382
$T19_{MM}$	51	47	25	1	12	76	90	21	39	17	11	390
$T20_{MM}$	45	47	22	2	12	72	94	20	39	17	11	381

Table 6.6: Number of incorrectly tagged part of speech

	A	C	D	I	J	N	$\sum$
# of part of speech in test data	3,914	1,439	1,818	7	1,847	10,203	19,228

Table 6.7: Number of particular part of speech classes in test data (adjectives (A), numerals (C), adverbs (D), interjections (I), conjunctions (J), nouns (N))

	P	R	T	V	X	$\sum$
# of part of speech in test data	2,200	3,378	143	4,278	745	10,744

Table 6.8: Number of particular part of speech classes in test data (pronouns (P), prepositions (R), particles (T), verbs (V), unknowns (X))

TAGGER	POS		SubPOS	g	n	c
$T0_{MM}$	362	1.21	1.31	**2.45**	**2.60**	**6.89**
$T1_{MM}$	383	1.28	1.38	**2.64**	**2.80**	**7.50**
$T2_{MM}$	376	1.25	1.37	**2.67**	**2.77**	**7.29**
$T3_{MM}$	371	1.24	1.35	**2.62**	**2.81**	**7.46**
$T4_{MM}$	370	1.23	1.35	**2.56**	**2.74**	**7.40**
$T5_{MM}$	377	1.26	1.36	**2.61**	**2.77**	**7.16**
$T6_{MM}$	365	1.22	1.32	**2.59**	**2.71**	**7.26**
$T7_{MM}$	379	1.26	1.38	**2.55**	**2.74**	**7.38**
$T8_{MM}$	385	1.28	1.39	**2.64**	**2.81**	**7.45**
$T9_{MM}$	370	1.23	1.34	**2.59**	**2.79**	**7.52**
$T10_{MM}$	374	1.25	1.35	**2.52**	**2.68**	**7.34**
$T11_{MM}$	391	1.30	1.42	**2.63**	**2.72**	**7.39**
$T12_{MM}$	381	1.27	1.37	**2.54**	**2.76**	**7.39**
$T13_{MM}$	362	1.21	1.32	**2.54**	**2.66**	**7.25**
$T14_{MM}$	378	1.26	1.37	**2.62**	**2.69**	**7.51**
$T15_{MM}$	376	1.25	1.35	**2.57**	**2.68**	**7.31**
$T16_{MM}$	382	1.27	1.39	**2.60**	**2.67**	**7.33**
$T17_{MM}$	384	1.28	1.39	**2.59**	**2.80**	**7.29**
$T18_{MM}$	382	1.27	1.39	**2.57**	**2.69**	**7.39**
$T19_{MM}$	390	1.30	1.40	**2.66**	**2.75**	**7.37**
$T20_{MM}$	381	1.27	1.38	**2.52**	**2.69**	**7.15**

Table 6.9: MM taggers: error rates (%) over particular morphological categories (part I)

TAGGER	possg	possn	p	t	d	a	v	s
$T0_{MM}$	0.04	0.01	0.21	0.22	0.29	0.66	0.22	0.15
$T1_{MM}$	0.03	0.00	0.25	0.25	0.31	0.70	0.25	0.15
$T2_{MM}$	0.04	0.01	0.24	0.24	0.29	0.68	0.24	0.15
$T3_{MM}$	0.04	0.01	0.24	0.24	0.30	0.68	0.24	0.15
$T4_{MM}$	0.04	0.01	0.24	0.24	0.29	0.67	0.24	0.15
$T5_{MM}$	0.04	0.01	0.24	0.23	0.31	0.67	0.23	0.13
$T6_{MM}$	0.04	0.01	0.22	0.24	0.31	0.67	0.24	0.15
$T7_{MM}$	0.04	0.01	0.26	0.26	0.31	0.68	0.26	0.15
$T8_{MM}$	0.04	0.01	0.25	0.26	0.32	0.68	0.26	0.15
$T9_{MM}$	0.04	0.01	0.24	0.24	0.30	0.67	0.24	0.13
$T10_{MM}$	0.04	0.01	0.22	0.21	0.31	0.66	0.21	0.15
$T11_{MM}$	0.05	0.02	0.26	0.25	0.32	0.71	0.25	0.15
$T12_{MM}$	0.04	0.01	0.23	0.23	0.32	0.68	0.23	0.15
$T13_{MM}$	0.04	0.01	0.22	0.23	0.30	0.67	0.23	0.16
$T14_{MM}$	0.04	0.01	0.24	0.24	0.30	0.68	0.24	0.15
$T15_{MM}$	0.03	0.00	0.23	0.26	0.31	0.67	0.26	0.15
$T16_{MM}$	0.04	0.01	0.24	0.26	0.32	0.70	0.26	0.15
$T17_{MM}$	0.05	0.02	0.24	0.25	0.31	0.71	0.25	0.15
$T18_{MM}$	0.04	0.01	0.24	0.24	0.33	0.70	0.24	0.15
$T19_{MM}$	0.04	0.01	0.24	0.25	0.31	0.68	0.25	0.15
$T20_{MM}$	0.04	0.01	0.25	0.26	0.31	0.67	0.26	0.15

Table 6.10: MM taggers: error rates (%) over particular morphological categories (part II)

# OF WORDS	N-TUPLES OF MISTAGGED MCs
10	POS SubPOS g c p t d a
12	POS SubPOS c p t d a
13	POS SubPOS g n c t d a
14	POS SubPOS g n c d
23	SubPOS d a
32	POS SubPOS
46	g n
57	POS SubPOS c
93	g n c
99	POS SubPOS g n c a
102	n
128	g c
232	g
332	n c
1189	c

Table 6.11: Mutual erroneousness of morphological categories in the test data tagged by the original MM tagger

For each word (in a position p) in the tagged test data, we construct on the tag level a sorted set $\mathbf{3T}_p$ of 3-tuples [n_i, t_i, p] where n_i is the number of taggers that posited the tag t_i for a word in the position p. The given set is in the order of n_i so that $n_1 \geq n_2 \geq n_3 \geq \ldots$ Be $\mathbf{T}_p$ ($|T_p| = |3T_p|$) set of all different tags posited by N taggers for a word in the position p. Then

$$\sum_{i=1}^{|T_p|} n_i = N.$$

Further, it is necessary also to consider the version of the test data tagged by the original tagger. That is why we construct a set of pairs [$\mathbf{t}_{0,p}$, **p**]. Each pair [$t_{0,p}$, p] provides an information saying that the original tagger disambiguates a word in the position p in the test data by the tag $t_{0,p}$. Consequently, for a word in the position p in the test data, there are two input arguments for the combination on the tag level: the set $3T_p$ and the pair [$t_{0,p}$, p].

For a combination on the subtag level, more input arguments come into play. Besides $3T_p$, [$t_{0,p}$, p], we work with a set $\mathbf{3ST}_p^i$ of 3-tuples [m_l, r_l, st_l] and a set $\mathbf{4ST}_p^i$ of 4-tuples [m_l, r_l, st_l, T_l], $m_l \leq N$, $r_l \leq |3T_p|$, $|T_l| \leq |3T_p|$. To create a set $3ST_p^i$, we use the information expressed in the set $3T_p$. Each set $3ST_p^i$ relates to a word in the position p and to a subtag in the position i (i goes from 1 to 15, see above). To find the elements (3-tuples) of a set $3ST_p^i$, we must look at tags in the set $3T_p$, determine all MC values in the position i, and keep the counts that say how many times (r_l) MC value st_l was observed in $3T_p$ and how many taggers (m_l) posit this value. Set $3ST_p^i$ = {[m_1, r_1, st_1],[m_2, r_2, st_2], ... } is generated first in the order of r_l so that $r_1 \geq r_2 \geq r_3 \geq \ldots$; if $r_m = r_n$ then the set is generated in order of m_l decreasingly.

The first three elements of each 4-tuple in a set $4ST_p^i$ are generated by the same process as the $3ST_p^i$ elements. In addition, there is one more element T_l for each 3-tuple [m_l, r_l, st_l]. For each value st_l of the *i-th* subtag, we are interested in tags which contain the value st_l in the *i-th* position. All these tags (numbered in the scope of $3ST_p^i$) are the elements of the set T_l.

For instance, 20 (N=20) bagged MM taggers return four different tags for the word "*co*" in the position 647 (p=647) in the test data, i.e.

T_{647} = {Db-------------, J,-------------, PQ--4----------, TT-------------}. Tag Db------------- is posited by 1 bagged MM tagger, tag J,------------- by two bagged MM taggers, tag PQ--4---------- by four bagged MM taggers and tag TT------------- by thirteen bagged MM taggers. Given this information, we get the set $3T_{647}$ = {[13, TT-------------, 647], [4, PQ--4----------, 647],[2, J,-

------------, 647], [1, Db-------------, 647]}. Obviously, 1+2+4+13=20. The original MM gives the pair [PQ--4---------, 647].

On the subtag level, $3ST^{1}_{647}$ = {[13,1,T],[4,1,P],[2,1,J],[1,1,D]}, $3ST^{2}_{647}$ = {[13,1,T],[4,1,Q],[2,1,,],[1,1,b]} and so on; $4ST^{3}_{647}$ = {[20,4,-,{t_1,t_2,t_3,t_4}]}, $4ST^{4}_{647}$ = {[20,4,-,{t_1,t_2,t_3,t_4}]}, $4ST^{5}_{647}$ = {[16,3,-,{t_1,t_3,t_4}], [4,1,4,{t_2}]}, etc.

6.4.1 Voting Strategies on the Tag Level

Algorithm No.1 - plurality voting (Fig. 6.1) The basic idea of the plurality (simple) voting is as natural as one would expect: let n be a number of different taggers which operate over the same text. Then, for the currently tagged word w_i, the taggers assign m different tags ($n \geq m$) Diff_Tags = $\{t_i^1, t_i^2, ..., t_i^m\}$. Consequently, for each element ($t_i^j, j = 1..m$) of Diff_Tags set we are able to count the number (c^j) of taggers which assigned to the given word the tag t_i^j. Then, we assign to the given word w_i such a tag t_i^k for which vote the plurality of the taggers, i.e. $c^k > c^j, j = 1...m$ and $j \neq k$. If there appear more such tags, then it is necessary to formulate a further criterion according to which we select just one tag. In case of five input taggers, there are totally seven different vote distributions - 5:0, 4:1, 3:2, 3:1:1, 2:1:1:1, 2:2:1, 1:1:1:1:1[4]. We can uniquely select the winning tag for the first five vote distributions. For the remaining two vote distributions (2:2:1, 1:1:1:1:1), we can apply a specific criterion: choose such tag candidate from a set of tag candidates as the winner[5] for which the best tagger within a set of taggers engaged in voting process votes; randomly choose a tag candidate for taggers with the same tagging performance.

To illustrate this given criterion, let us suppose (we assume the ratio 2:2:1) that for a tag t_i^j the tagger with tagging accuracy 91% and tagger with accuracy 92% vote; for a tag t_i^k tagger with tagging accuracy 89.5% and tagger with accuracy 90% vote. Then, since the tagger with tagging accuracy 92% is the best one (within a set of voting taggers), the tag t_i^j is the winner of voting.

According to the way we define the set $3T_p$, the most “popular” tag is an element of the first 3-tuple.

[4]Using the formalism desribed above, we mention the corresponding set of different tags returned by five taggers and the corresponding number of votes for each ratio: 5:0 – Diff_Tags = $\{t_i^1\}$, $c^1 = 5$; 4:1 – Diff_Tags = $\{t_i^1, t_i^2\}$, $c^1 = 4$, $c^2 = 1$; 3:2 – Diff_Tags = $\{t_i^1, t_i^2\}$, $c^1 = 3$, $c^2 = 2$; 3:1:1 – Diff_Tags = $\{t_i^1, t_i^2, t_i^3\}$, $c^1 = 3$, $c^2 = 1$, $c^3 = 1$; 2:1:1:1 – Diff_Tags = $\{t_i^1, t_i^2, t_i^3, t_i^4\}$, $c^1 = 2$, $c^2 = 1$, $c^3 = 1$, $c^4 = 1$; 2:2:1 – Diff_Tags = $\{t_i^1, t_i^2, t_i^3\}$, $c^1 = 2$, $c^2 = 2$, $c^2 = 1$; 1:1:1:1:1 – Diff_Tags = $\{t_i^1, t_i^2, t_i^3, t_i^4, t_i^5\}$, $c^1 = 1$, $c^2 = 1$, $c^3 = 1$, $c^4 = 1$, $c^5 = 1$.

[5]There exist two candidates for the ratio 2:2:1 and five candidates for the ratio 1:1:1:1:1.

```
procedure Plurality_Voting(3T_p)
begin
  return(t_1);
end
```

Figure 6.1: Plurality voting

Algorithm No.2 - plurality voting driven by the original tagger (Fig. 6.2) As the performance of the original tagger is higher than the performance of the bagged taggers, it is reasonable to include the original tagger into the combination step as well. On the other hand, the most "popular" tag need not to be the right one. One way to select a tag that was not chosen by a plurality of taggers is to give preference to the original tagged output because of its highest performance. Thus, if a tag in the second 3-tuple in $3T_p$ is the same as the tag which the original tagger gives we prefer it.

```
procedure Plurality_Voting_Driven_T0_HMM(3T_p, [t_0,p,p]);
begin
  if n_1 == n_2 then return(t_1)
                 else if t_2 == t_0,p then return(t_2)
                                      else return(t_1);
                    fi
  fi
end
```

Figure 6.2: Plurality voting driven by the original tagger

Algorithm No.3 - plurality voting driven by the original tagger and by the parameter C (Fig. 6.3) We modify slightly the idea of plurality voting driven by the original tagger in the sense of tuning the decisions based on the plurality votes according to a parameter C. The parameter C goes from 1 to N (N is equal to the number of input taggers, see above); if the number of plurality votes for a given tag is greater than C we prefer the most "popular" tag without regard to the tag posited by the original tagger. Else, we follow the idea of the algorithm No. 2.

6.4.2 Voting Strategies on the Subtag Level

Algorithm No.4 - plurality voting on the subtag level (Fig. 6.4) First, we do not take into account the truths about *MC*s `case`, `number` and `gender`. In addition, we tag each subtag independently without regard to

```
procedure Plurality_Voting_Driven_T0_HMM_C(3T_p, [t_{0,p},p], C);
begin
  if n_1 > C then return(t_1)
             else
                  if t_2 == t_{0,p} then return(t_2)
                                    else return(t_1);
                  fi
  fi
end
```

Figure 6.3: Plurality voting driven by the original tagger and a parameter C

the subtag context. The assignment of a value[6] *st* to the *i-th* subtag variable is driven by two criteria: (a) how many times a value *st* appears in the set $3T_p$ (numbers r_j in $3ST_p^i$); (b) how many taggers posit a value *st* (numbers m_j in $3ST_p^i$); if we cannot decide according to the criterion (a) which value is selected, then we use the criterion (b). In compliance with the way in which the set $3ST_p^i$ is constructed, it is not difficult to find the "winning" value. Unfortunately, the given strategy of plurality voting applied on subtags can lead to a selection of a meaningless tag.

```
procedure Plurality_Voting_Subtag_Level();
begin
  FTag := "";
  for i = 1 to = 15 do
      FTag := FTag * Eval(3ST_p^i);
  od
  return(FTag);
  proc Eval(3ST_p^i) ≡
    return(st_1).
end
```

Figure 6.4: Plurality voting on subtag level

Algorithm No. 5 (Fig. 6.5) The prior analyses have shown as problematic the *MC*s `case`, `number` and `gender`, which we now tag in a way conditioned by mutual dependence. The remaining *MC*s are tagged according to the algorithm No. 4. By the algorithm No.4 we find the output `case`

[6]The list of all potential values of *i-th* subtag is determined by 3-tuple members st_j in the set $3ST_p^i$.

value[7] st_n^5. Next, we know tags (T_n^5) which have in the 5th position the value st_n^5. The selection of the output `number` value st_m^4 is driven by the algorithm No. 4 as well and the set of all potential values is limited to the values not only in the tags with the given value, but also in the tags from the set T_n^5; in other words, members of $T_n^5 \cap T_m^4$. Finally, the output `gender` value st_k^3 is determined by algorithm No. 4 and the intersection of sets T_n^5, T_m^4 and T_k^3. Using this strategy, at least the combinations of `case`, `number` and `gender` values make sense, i.e. corresponds to the output of MA.

Let us take an example from our test data to illustrate the core ideas of the algorithms No. 4 and No. 5 more explicitly. The word "*srovnatelné*" in the position 11 annotated as **AAIP1----A1----** is totally ($N = 20$) tagged by three different tags. Then, the set $3\mathrm{T}_{11}$ contains three elements: $3\mathrm{T}_{11}$ = {[17, AANS1----A1----, 11], [2, AAIP1----A1----, 11], [1, AAFP1----A1----, 11]}. We observe a trade-off between a neuter (N), a masculine inanimate (I) and a feminine (F) adjective in the position corresponding to a gender (3rd position), and a singular (S) and a plural (P) adjective in the 4th position. Sets $3\mathrm{ST}_{11}^i$ covering this kind of information contain the following elements: $3\mathrm{ST}_{11}^3$ = {[17,1,N],[2,1,I],[1,1,F]}, $3\mathrm{ST}_{11}^4$ = {[3,2,P],[17,1,S]}. Similarly, sets $4\mathrm{ST}_{11}^3$ = {[17,1,N,{t_1}],[2,1,I,{t_2}],[1,1,F,{t_3}]}, $4\mathrm{ST}_{11}^4$ = {[3,2,P,{t_2, t_3}],[17,1,S,{t_1}]}. In addition, $3\mathrm{ST}_{11}^5$ = {[20,3,1]}, $4\mathrm{ST}_{11}^5$ = {[20,3, 1, {t_1, t_2, t_3}]}. Let us first trace the procedure *Plurality_Voting_Subtag_Level*. The individual steps are expressed in Table 6.11.

Then, we trace the procedure *Plurality_Voting_Subtag_Level_cgn* only for $i = 3,4,5$. For the other steps, see Table 6.12.

Algorithm No. 4 tags the input word by the tag **AANP1----A1----** and algorithm No. 5 by the tag **AAIP1----A1----**, which is the correct one.

6.5 Discussion of the Results

Discussing the results we obtained we should answer the following questions:

1. *Why bagging?* Bagging is effective in cases where small variations in the training data result in significant differences in the resulting classifiers. If training is relatively insensitive to small training data differences, then the N resulting classifiers will not be significantly different, and therefore combining these classifiers will not give any significant improvement over a single classifier. For our first bagging

[7] To distinct `case`, `number` and `gender` 3/4-tuples, we add as the exponent numbers 5, 4, and 3, respectively.

i	r_1	r_2	k	OUTPUT VALUE
1	3			A
2	3			A
3	1	1	1	N
4	2	1		P
5	3			1
6	3			-
7	3			-
8	3			-
9	3			-
10	3			A
11	3			1
12	3			-
13	3			-
14	3			-
15	3			-

Table 6.12: Procedure *Plurality_Voting_Subtag_Level* - example

i	r_1	T_1	$T \cap T_1$	r_2	T_2	$T \cap T_2$	OUTPUT VALUE	T
5	3	$\{t_1,t_2,t_3\}$	$\{t_1,t_2,t_3\}$				1	$\{t_1,t_2,t_3\}$
4	2	$\{t_2,t_3\}$	$\{t_2,t_3\}$	1	$\{t_1\}$	$\emptyset$	P	$\{t_2,t_3\}$
3	1	$\{t_1\}$	$\emptyset$	1	$\{t_2\}$	$\{t_2\}$	I	$\{t_2\}$

Table 6.13: Procedure *Plurality_Voting_Subtag_Level_cgn* - example

experiment on Czech, we took as the original classifier the Collins parser retrained for Czech (Sect. 5.2.2). The positive results obtained have provided us the hope for the bagging experiment on Czech with the MM tagger as the original classifier.

In order to compare the results obtained with Czech bagging experiments applied on the problem of parsing, we consider the union of all pairs [tag, word position in test data] given by the output of the 20 bagged taggers applied on the test data and keep all tags posited by at least one bagged tagger. We get the number of correctly tagged words by summing the "*correct*" numbers in the column marked "correct/incorrect" in Table 6.15 with regard to the number of taggers that posited the given pairs. For example, 18 bagged taggers agree on 450 tags from which 289 are assigned correctly and 161 assigned incorrectly. As the number of bagged taggers the output tags of which agree decreases (the second column), tagging accuracy on corresponding tags decreases as well (the last column). The unbalanced output is created by keeping all pairs that are posited by more than the half of the bagged classifiers (see above). As we work with 20 bagged taggers, the precision of unbalanced output is 91.80%. Keeping the pairs posited by 20 bagged taggers, we get precision 95.96% and the pairs posited by at least one tagger have the precision 83.80% (see Tab. 6.14).

2. *(In)Dependence of bagged taggers?* According to the motivation given in Sect. 5.1, only a combination of the independent classifiers working with the accuracy greater than 50% can be successful. The Complementary Rate (CR, Sect. 5.3) represents a quantitative measure of the taggers (in)dependence. All the input taggers (original and bagged) are dependent (Tab. 6.3 and Tab. 6.4); the CRs(original tagger, bagged tagger)s are lower than the CRs(bagged tagger, bagged tagger). The bagged tagger CRs lie within the limits 15%-20%, i.e. on average, every sixth word token is tagged by two bagged taggers differently. On the contrary, the CRs(original tagger, bagged tagger) lie within the limits 9%-13%, i.e. on average, every ninth word token is tagged by the original and bagged taggers differently. Since the average independence of the bagged parsers on Czech was 7.1% (the percentage of dependencies that were output by Parser 1 and not by Parser 2), which is less than in our case, we could expect an improvement through the voting of the taggers (since the voting of bagged and original parsers achieved an improvement).

3. *Why no improvement?* The balanced combination methods are represented by algorithms No.1 - No.5. The tagging accuracies of these methods (Tables 6.16, 6.17) do not reach the tagging accuracy of the original tagger. In detail, the results of the tag level algorithms (Table 6.16) are better than the results of the subtag level algorithms. As the MM tagger works with tags on the tag level, it is understable that the strategy of the subtag level algorithms is slightly different in comparison with the MM strategy. All chosen tag level strategies or subtag level strategies are affected by the lack of the tag/subtag context information. In algorithm No.5, we attempt at least partially to include the context information into the combination. We split the set of subtags into 13 subtag subsets {`POS`},{`SubPOS`},{`g,n,c`}, {`possg`}, {`possn`},{`p`},{`t`},{`d`},{`a`},{`v`}, {`x1`}, {`x2`} and {`s`}. Each subset was processed independently on the others and without any context information. Only the processing of the subset {`g,n,c`} takes at least partially the advantage of the subtag context.

   We see that the unbalanced method gives significantly better results than the balanced method. But neither the unbalanced nor the balanced method achieves an improvement over the original tagger in comparison with the successful bagging on parsing.

   Summing up the basic bagging parameters, we discuss the following numbers:

   - number of bagged parsers/taggers: 18 vs. 20
   - average dependence of bagged parsers/taggers: 7.1% vs 17.5%
   - the level of performance of bagged parsers/taggers: 76% vs. 90%
   - number of unique dependencies/tags output by bagged parsers/taggers: 100K vs. 34K

   Given the first three parameters, we can hope for the success of the bagging on tagging. But, there were approximately 100K unique dependencies output by the 18 bagged parsers. Approximately, 40% of these dependencies were output by all 18 bagged parsers; for the bagged taggers, there were approximately 34K of unique tags output by the 20 bagged taggers and approximately 78% of these tags were output by all 20 bagged taggers. The "weak" robustness of the bagged tagger system (34K vs. 100K, 78% vs. 40%) is probably the reason why the bagging on tagging does not work so well as on the parsing.

# OF TAGS POSITED AT LEAST X TAGGERS	PRECISION(%)	RECALL(%)	F-MEASURE(%)
$x = 1$	83.80	95.29	89.18
$x = 11$	91.80	91.12	91.45
$x = 20$	95.96	85.00	90.15

Table 6.14: Accuracy of combined bagged taggers

# OF TAGGERS	# OF POSITED TAGS	CORRECT/ INCORRECT	TAGGING ACCURACY (%)
20	26,551	25,476/1,075	95.95
19	661	454/207	68.66
18	450	289/161	64.22
17	359	211/148	58.77
16	348	196/152	56.32
15	366	199/167	54.37
14	284	144/140	50.70
13	274	138/136	50.36
12	226	95/131	42.04
11	226	107/119	47.35
10	267	114/153	42.70
9	229	88/141	38.43
8	237	99/138	37.08
7	300	111/189	37.00
6	317	114/203	35.96
5	391	127/264	32.49
4	396	109/287	27.53
3	488	140/348	28.69
2	611	152/459	24.88
1	1,010	197/813	19.50

Table 6.15: Tagging accuracy versus the number of MM taggers positing a tag

TAGGER/ STRATEGY	PARAMETERS	# OF TAG LEVEL ERRORS	#OF SUBTAG LEVEL ERRORS	TAGGING ACCURACY (%)
$T0_{MM}$	-	2,540	4,872	91.53
$T20_{MM}$	-	2,620	5,084	91.26
Alg. No.1	N=20 $T1_{MM}$-$T20_{MM}$	2,573	4,972	91.42
Alg. No.1	N=21 $T0_{MM}$-$T20_{MM}$	2,569	4,963	91.43
Oracle	$T1_{MM}$-$T20_{MM}$	-	-	95.29
Oracle	$T0_{MM}$-$T20_{MM}$	-	-	95.29
Alg. No.2	-	2,570	4,929	91.41
Alg. No.3	C=9	2,573	4,971	91.41
Alg. No.3	C=10	2,570	4,968	91.43
Alg. No.3	C=11	2,567	4,958	91.44
Alg. No.3	C=12	2,572	4,949	91.42
Alg. No.3	C=13	2,572	4,944	91.42
Alg. No.3	C=14	2,569	4,932	91.43
Alg. No.3	C=15	2,575	4,936	91.41

Table 6.16: Results of the tag level algorithms

TAGGER/ STRATEGY	PARAMETERS	# OF TAG LEVEL ERRORS	#OF SUBTAG LEVEL ERRORS	TAGGING ACCURACY (%)
Alg. No.4	N=20 $T1_{MM}$-$T20_{MM}$	2,611	4,954	91.29
Alg. No.4	N=21 $T0_{MM}$-$T20_{MM}$	2,600	4,929	91.33
Alg. No.5	N=20 $T1_{MM}$-$T20_{MM}$	2,600	4,966	91.33
Alg. No.5	N=21 $T0_{MM}$-$T20_{MM}$	2,588	4,939	91.37

Table 6.17: Results of the subtag level algorithms

```
procedure Plurality_Voting_Subtag_Level_cgn();
i = 1, T : set of int;
begin
   FTag = "";
   while i <= 15 do
           if i == 3 then
                             FTag = FTag * Eval_cgn(4ST_p^3, 4ST_p^4, 4ST_p^5);
                             i+ = 2;
                     else
                             FTag = FTag * Eval(3ST_p^i);
           fi
           i + +;
   od
   return(FTag);
   proc Eval_cgn(4ST_p^3, 4ST_p^4, 4ST_p^5) ≡
      int c, g, n;
      c = Eval_MC_i_j(3ST_p^5, 1);
      T = T_c^5;
      FSTags = st_c^5;
      n = Eval_MC_i(4ST_p^4);
      FSTags = st_n^4 * FSTags;
      g = Eval_MC_i(4ST_p^3);
      FSTags = st_g^3 * FSTags;
      return(FSTags).
   proc Eval_MC_i_j(3ST_p^i, j) ≡
      return(j).
   proc Eval_MC_i(4ST_p^i) ≡
      empty = 1, j = 1;
      while (not empty) do
              if ||T_j ∩ T|| == 0 then j + +
                                  else empty = 0;
              fi
      od
      T = T ∩ T_j;
      return(Eval_MC_i_j(3ST_p^i, j)).
end
```

Figure 6.5: Plurality voting on subtag level employing context information

Chapter 7

Original Combination of Czech Taggers

Motivated by the experiments with the combination of original English taggers, we performed an experiment of the simple (plurality) voting combination of the original Czech taggers trained on the Czech Tagged Corpus (CTC) and the Prague Dependency Treebank (PDT).

7.1 Original Taggers Trained on CTC

Tab. 3.29 in Chapter 3 above shows that the taggers trained on CTC based on the different tagging strategy have comparable results (except the Hlinsko experiment). Tab. 7.1 presents concrete values of the complementary rates of the mentioned taggers. Naturally, as the Prague experiment achieves the best results relative to the other experiments the CRs(Prague,[Hlinsko|Mariánská|Baltimore|Washington]) are smaller than the other CRs. On the other hand, the Hlinsko experiment gives the worst results, i.e. produces the highest number of errors and CRs(Hlinsko,[Prague|Mariánská|Baltimore|Washington]) are the highest ones.

We first included five (Hlinsko, Prague, Mariánská, Baltimore, Washington) taggers into a simple voting procedure[1]. We expected no magic results especially because of the lower quality of the Hlinsko experiment. Then, we

[1]Our voting strategy is strictly directed by the criterion specified in Sect. 6.4.1.

	Hlinsko	Prague	Mariánská	Baltimore	Washington
Hlinsko	0	52.24	50.90	49.55	43.05
Prague	11.62	0	12.03	26.56	22.41
Mariánská	11.34	14.47	0	28.74	25.91
Baltimore	16.97	34.69	35.06	0	10.70
Washington	14.19	36.82	38.18	18.24	0

Table 7.1: Complementary rates (%) of original Czech taggers trained on CTC

excluded the less “productive” taggers Hlinsko, Mariánská, Washington. As the Prague tagger gives results with the highest tagging accuracy (81.38%) (we take it as the baseline tagger) and the tagging accuracy of the combination reaches 81.99%, we achieve the improvement of 0.61% over the baseline tagger. Detailed analysis of vote distributions (Tab. 7.3) allowed us to take the next step in direction of better overall accuracy. For the word forms with the vote distribution 1:1:1, the rule-based tagger Baltimore gives the best accuracy; thus, we prefer a tag determined by the Baltimore tagger in case of the 1:1:1 vote distributon.

7.2 Original Taggers Trained on PDT

Similarly to the taggers trained on CTC, the taggers trained on PDT (the experiments EXP_CZ, MM_CZ$_{tri}$ and MM_CZ$_{bi}$, see Tab. 3.29) present mutually comparable results; even, the complementary rates (Tab. 7.2) are higher than the ones presented in Tab. 7.1. Despite the higher tagging accuracies and the higher complementary rates, the plurality voting of the EXP and MM taggers did not bring about non-zero a positive change of the overall tagging accuracy. Thus, we analyzed the quantitative statistics of vote distributions in detail in Tab. 7.3. Most of all, we concentrated on the cases of the 2:1 vote distribution most of all. Taking into the consideration the different methodologies of the input taggers it is no wonder that the 2:1 vote distributions MM taggers vs. EXP or RB tagger are the most frequent out of all possible 2:1 vote distributions. For instance, 2.141 word forms were tagged (1.249 word forms incorrectly, 892 word forms correctly) by the vote distribution MM_CZ$_{tri}$, MM_CZ$_{bi}$ (same tag) vs. EXP_CZ (different tag). The significant success of the voting Prague, Mariánská:Baltimore in comparison with the failure of the voting MM_CZ$_{tri}$, MM_CZ$_{bi}$: EXP_CZ is probably caused by the higher accuracy of the taggers Prague, Mariánská over the tagger Baltimore in comparison with the higher accuracy of the EXP_CZ over the taggers MM_CZ$_{tri}$, MM_CZ$_{bi}$. To take advantage of the high complementary rates of the taggers trained on the PDT, we should use the context-based combination method; i.e. we have to locate the contexts within which the EXP tagger works accurately than the MM taggers and on the other hand, the context within which the MM taggers works more accurately than EXP tagger.

	EXP_CZ	MM_CZ$_{tri}$	MM_CZ$_{bi}$
EXP_CZ	0	47.09	41.58
MM_CZ$_{tri}$	50.91	0	12.13
MM_CZ$_{bi}$	52.19	22. 50	0

Table 7.2: Complementary rates (%) of original Czech taggers trained on PDT

corpus		CTC	PDT
test file # of word tokens		1.294	29.972
vote distributions	3:0	1.014	26.721
	correct	939	25.892
	incorrect	75	829
	2:1	225	3.073
	correct	114[a]	1.542[b]
	incorrect	111[c]	1.531[d]
	1:1:1	55	178
	correct	25[e]	141[f]
	incorrect	30	37
TA(oracle)		87.48	96.13

[a] Mariánská, Baltimore:Prague 14, Prague, Mariánská:Baltimore 79, Prague, Baltimore:Mariánská 21

[b] MM_CZ$_{tri}$, MM_CZ$_{bi}$:EXP_CZ 892, MM_CZ$_{tri}$, EXP_CZ:MM_CZ$_{bi}$ 430, EXP_CZ, MM_CZ$_{bi}$:MM_CZ$_{tri}$ 220

[c] Mariánská, Baltimore:Prague 12, Prague, Mariánská:Baltimore 75, Prague, Baltimore:Mariánská 24

[d] MM_CZ$_{tri}$, MM_CZ$_{bi}$:EXP_CZ 1.249, MM_CZ$_{tri}$, EXP_CZ:MM_CZ$_{bi}$ 101, EXP_CZ, MM_CZ$_{bi}$:MM_CZ$_{tri}$ 181

[e] Prague 7, Mariánská 1, Baltimore 17

[f] EXP_CZ 62, MM_CZ*tri* 63, MM_CZ$_{bi}$ 16

Table 7.3: The vote distributions

Chapter 8

More "PDT-like" Language Resources

Czech Tagged Corpus (CTC, Sect. 3.1.2)) is an annotated corpus containing about 600K word tokens. The format of CTC is very simple: each word together with its CTC tag occupies one separate line (see Tab. 8.1). As 600K word tokens are not a negligible amount of morphologically annotated texts, we have decided to convert CTC into the Prague Dependency Treebank (PDT) format (which is SGML-based, words are annotated by the positional tags and the inner format corresponds to SGML coding). To get a "new coat" of the CTC, we have to undertake several steps during which we obtain various intermediate corpora:

CTC^{words}	# only words from the CTC stripped of their CTC tags
CTC^{words}_{pdt}	# CTC^{words} in the SGML coding
CTC^{mm}_{pdt}	# CTC$^{words}_{pdt}$ morphologically analyzed
CTC^{pos}	# the CTC in original format, the tags of which
	# are mapped into the "new" positional tags
CTC^{aut}_{pdt}	# CTC^{pos} automatically converted in SGML format
CTC_{pdt}	# CTC in the PDT format - final version

Tables 8.1, 8.2 and 8.3 provide the samples from the particular corpora using the example

...	budu	vyjadřovat	v	decibelech	stav	...
...	(I) will express		in decibels		state	...

8.1 Mapping the CTC Tags into the Positional Tags

The CTC tag set was designed without paying too much attention to AMA because the procedure of AMA had not been completed at that time. As we plan to include the CTC into the PDT, the tags must correspond to the positional tag system of the MA. Table 8.4 shows a mapping from the CTC tags into the positional tags. As the number of MCs included in the positional tag system is higher than that in the CTC tag system, we "fill in" just positions (values) corresponding to the known MCs from the CTC tagset. The meaning of particular MC variables is explained in Tab. 3.1.

CTC		CTC^{words}	$\mathrm{CTC}_{pdt}^{words}$
budu	X	budu	<f>budu
vyjadřovat	VTA	vyjadřovat	<f>vyjadřovat
v	Rv	v	<f>v
decibelech	NIP6	decibelech	<f>decibelech
stav	NIS4	stav	<f>stav

Table 8.1: The samples from the corpora CTC, CTC^{words} and $\mathrm{CTC}_{pdt}^{words}$

CTC_{pdt}^{mm}	CTC^{pos}	
<f>budu<MMl>být<MMt>VB-S---1F-AA---	budu	XX-------------
<f>vyjadřovat<MMl>vyjadřovat<MMt>Vf--------A----	vyjadřovat	VB-S---1F-AA---
<f>v<MMl>v<MMt>RR--4----------<MMt>RR--6----------	v	RR-------------
<f>decibelech<MMl>decibel<MMt>NNIP6-----A----	decibelech	NNIP6----------
<f>stav<MMl>stav[a] <MMt>NNIS1-----A----	stav	NNIS4----------
<MMt>NNIS4-----A----		
<MMl>stavět[b] <MMt>Vi-S---2--A----		
<MMl>stavit[c] <MMt>Vi-S---2--A----		
<MMl>stát[d] <MMt>VmYS------A----		

[a] a state - in nominative, accusative case
[b] to build, to construct - imperative form
[c] to drop in - imperative form
[d] to betide - archaic present transgressive of perfective verb

Table 8.2: The samples from the corpora CTC^{pos} and CTC_{pdt}^{mm}

CTC_{pdt}^{aut}	CTC_{pdt}
<f>budu<l>být<t>VB-S---1F-AA---	<f>budu<l>být<t>VB-S---1F-AA---
<f>vyjadřovat<l>vyjadřovat<t>Vf--------A----	<f>vyjadřovat<l>vyjadřovat<t>Vf--------A----
<f>v<MMl>v<MMt>RR--4----------	<f>v<l>v<t>RR--6----------
<MMt>RR--6----------	
<f>decibelech<l>decibel<t>NNIP6-----A----	<f>decibelech<l>decibel<t>NNIP6-----A----
<f>stav<l>stav<t>NNIS4-----A----	<f>stav<l>stav<t>NNIS4-----A----

Table 8.3: The samples from the corpora CTC_{pdt}^{aut} and CTC_{pdt}

CTC TAG	POSITIONAL TAG	DESCRIPTION
N*gnc*	NN*gnc*----------	noun
NZ	Xx-------------	abbreviation
A*gncda*	AA*gnc*----*da*----	adjective regular;
VT*a*	Vf--------*a*----	verb: infinitive
VW*n*P*sga*	Ve*gn*------*a*----	verb: transgressive present
VW*n*M*sga*	Vm*gn*------*a*----	verb: transgressive past
V*pn*AM*mga*	Vp*gn*---XR-*a*A---	verb: past participle, active
V*pn*PM*mga*	Vs*gn*---XX-*a*P---	verb: past participle, passive
V*pns*P*mga*	VB-*n*---*p*P-*a*A---	verb: indicative, present tense
V*pns*F*mga*	VB-*n*---*p*F-*a*A---	verb: indicative, future tense
V*pnst*R*ga*	Vi-*n*---*p*–*a*----	verb: imperative
PP*fnc*	PP-*nc*--*f*-------	personal pronoun I/you
PP3*gnc*	PP*gnc*--3-------	personal pronoun he/she/it
PR$g_1n_1c3g_2n_2$	PS$g_1n_1cg_2n_2$3-------	pronoun possesive 3rd person
PR$g_1n_1cfg_2n_2$	PSg_1n_1c-n_2f-------	pronoun possesive 1st, 2nd person
PS*gnc*	P8*gnc*----------	reflexive possesive pronoun
PE*c*	P7-X*c*----------	pronoun reflexive *se*
PD*gnca*	PD*gnc*----------	pronoun demonstrative
O*da*	Dg-------*da*----	adverb
SS	J,-------------	conjunction (subordinating)
SP	J^-------------	conjunction (coordinating)
C*gnc*	Cl*gnc*----------	numeral, basic
R*preposition*	RR-------------	preposition
F	II-------------	interjection
K	TT-------------	particle
T_SB	Z#-------------	sentence boundary
T_IP	Z:-------------	punctuation
X	XX-------------	unknown

Table 8.4: Mapping CTC tags into the positional tags

8.2 From CTC to CTC$_{pdt}$

A simple mapping of CTC tags into the positional tags is not sufficient, so that we have to check positional tags in the CTC^{pos} relative to the output of the AMA - CTC$^{mm}_{pdt}$. A *5-step procedure* was formulated to merge and to process the information coming from the corpora CTC^{pos} and CTC$^{mm}_{pdt}$ in order to get the final corpus CTC$_{pdt}$. The steps can be characterized as follows:

1. *Processing of morphologically unambiguous words in* CTC^{mm}_{pdt}. The words with unambiguous morphological analysis are annotated by this procedure. We do not take the CTC$_{pos}$ annotation into account.

2. *Processing of morphologically ambiguous words in* CTC^{mm}_{pdt}. To express the closeness (or identity) of two tags, we decided to define a distance metric. The most straightforward distance metric is the one given in equation (8.1), where A and B are the tags to be compared, and $\delta(a_i, b_i)$ is the distance between the values of the i-th subtag in a tag with n subtags (in our case $n = 15$).

$$D(A,B) = \begin{cases} \sum_{i=1}^{n} \delta(a_i, b_i) & \delta(a_1, b_1) > -1 \\ -1 & \delta(a_1, b_1) = -1 \end{cases} \tag{8.1}$$

   Distance between two subtags is measured using equation(8.2):

$$\delta(a_i, b_i) = \begin{cases} 0 & \mathrm{a}_i = \mathrm{b}_i, \\ 1 & \mathrm{a}_i \neq b_i,\ \mathrm{i} = 2 \ldots \mathrm{n} \\ -1 & \mathrm{a}_i \neq b_i,\ \mathrm{i} = 1 \end{cases} \tag{8.2}$$

3. *Evaluation of ambiguities in* CTC^{aut}_{pdt}.

4. *Manual and automatic resolution of ambiguities in* CTC^{aut}_{pdt}.

5. *Evaluation of manual and automatic resolution of ambiguities.*

Given the input example, we illustrate each step separately:

Step 1 For the unambiguous word *budu, vyjadřovat, decibelech* the AMA provides just a single tag (see the first column in the Table 8.2). The SGML markup <MMl> and <MMt> for lemmas and tags produced by AMA becomes the SGML markup <l> and <t> for annotated lemma and tag (see the second column in the Table 8.3).

Step 2 For the ambiguous word *v* the AMA provides two different tags and for the word *stav* five different tags. Let A denote the tag from CTC^{pos} and B denote the tag from CTC_{pdt}^{mm}. To express the closeness of two tags, we distinguish the equality of part of speech values. If the part of speech values are identical, the chosen metric D(A,B) expresses the number of different MC values in A and B. Otherwise we assume that A and B are totally different ($D(A, B) = -1$). We select from the list of tags provided by AMA such tag which has the minimal distance from the corresponding CTC tag. If there are more tags with the same minimal distance we keep all of them.

lemma		
v	*D*(RR-------------, RR--4----------)	= 1
	D(RR-------------, RR--6----------)	= 1
stav	*D*(NNIS4----------, NNIS1-----A----)	= 2
	D(NNIS4----------, NNIS4-----A----)	= 1
stavět	*D*(NNIS4----------, Vi-S---2--A----)	= -1
stavit	*D*(NNIS4----------, Vi-S---2--A----)	= -1
stát	*D*(NNIS4----------, VmYS------A----)	= -1

Step 3 After the 2nd step the word *v* remains ambiguous. The explanation is clear - the prepositions are annotated as "R" followed by the particular preposition (e.g. "Rv") in the course of CTC. According to the mapping strategy (Tab. 8.4), each preposition is mapped into the tag RR------------- . We cannot deduce the `case` information directly from the prepositional CTC tags. Thus, we can conclude that all prepositional tags provided by AMA have the same distance from the CTC tag of the given word and we cannot resolve the prepositional ambiguity in the 2nd step.

Totally, 60% of words in the corpus CTC_{pdt}^{mm} are ambiguous. Passing through the 2nd step we decrease the percentage of ambiguous words to 9.7% out of which 51% are ambiguous prepositions.

Step 4 The ambiguities which remain in the CTC_{pdt}^{aut} are resolved by two simultaneous procedures - manual and automatic. We formulate a list of templates for automatic resolution, for instance the *prepositional template*:

`if` NextWord is NOUN or ADJECTIVE or PRONOUN or NUMERAL `then` PrepositionCase = NextWordCase.

Step 5 The manual resolution eliminates the ambiguous words totally. On the other hand, we are not able to cover all ambiguous cases in templates. Comparing the manual and automatic resolution of those ambiguities on which the templates are aimed, we can improve the quality of manual disambiguation.

We illustrate the step 5 on the text originally containing 1,458 ambiguous words (out of which 815 ambiguous prepositions). The automatic template

`if` NextWordCase $\in \{1,2,3,4,5,6,7\}$ `then` PrepositionCase = Next Word Case

concentrates only on the prepositional ambiguities. Using this template we decrease the number of ambiguous words to 727 (i.e. we resolve 731 prepositional ambiguities); 84 prepositions remain ambiguous and 9 prepositions are disambiguated differently by automatic and manual procedures. Using the given prepositional template we cannot resolve the situations when the preposition is followed by an ambiguous word or by a non-ambiguous word with X-value of `case`. The following figure gives the idea of discrepancies between manual and automatic resolution of prepositional ambiguities.

errors	times
incorrect manual disambiguation	4
prepositional template doesn't express the given sentence situation	3
incorrect tagging of CTC	1
spelling error	1

The revision of manual resolution with regard to the evaluation of the given discrepancies provides final corpus CTC_{pdt}.

Chapter 9

Conclusions

Corpus-based approaches (as one of the corpus linguistics' topics) provide a new way to use information coming from the primary language resources - texts or speeches.

This thesis is aimed at a corpus-based solution of Czech tagging (using supervised training). The presence of a Czech annotated corpus[1] and the experience learned through the training of taggers operating on English have encouraged us to find out how useful are the corpus-based approaches for the task of Czech language tagging. From a historical point of view, the Czech tagging experiments are the first Czech (even Slavic language) corpus-based experiments of any kind. In our research, we had to carry out first an extensive data preparation stage before we reached the experimental stage.

The data preparation stage has a very specific position - it is really needed[2], it is very time-consuming, but nobody asks about the details[3]. The crucial thing is that we needed the total amount of morphologically annotated Czech data to be comparable with the size of the English data on which comparable experiment have been carried, namely the corpus referred to as the Wall Street Journal.

In *the experiment stage*, thanks to the morphologically annotated Czech corpora, we could dive into the corpus-based tagging experiments. During the first series of experiments, we have applied the Markov model and rule-based strategies to Czech. We should keep in mind that these tagging strategies are language-independent; the only language-dependent factor is the training corpus. As we were changing the tag set from the most detailed (thousands of tags) into a set containing mainly the part of speech information (tens of tags) the results were getting closer and closer to the results for English[4]. Taking into account only the tagging accuracy criterion, the best Czech result reached the level of 96%. However, the post-tagging applications may recquire not only the part of speech information, but may

[1] Albeit not directly suitable for our methods.
[2] Since we have used supervised training methods
[3] We refer to the process of annotation, not the theoretical background.
[4] All tagging experiments performed on English work with 96-97% tagging accuracy.

need a more detailed morphological information. Therefore, we could not stop here, we had to try to improve the results on the full tag set, which were below 94% both for the Markov model as well as the Exponential tagger. The error analysis suggested that the Markov model and Exponential taggers have the character of partially complementary classifiers.

The tagging strategies used are based on different algorithms. Are the errors produced by the taggers different as well? If yes, there is a chance to get better results by a combination of the taggers' output. Using the plurality voting as a combination method of the rule-based and Markov model taggers trained on the CTC, we got better results going beyond the point 82.69 % (81.38% $\rightarrow$ 82.69 %). Along with this improvement, we should remark that the improvement on this level of accuracy is not as significant as we have noticed when doubling the training corpus. The combination of the Exponential and Markov model taggers trained on the Prague Dependency Treebank by means of the plurality voting strategy did not bring any gain over the baseline Exponential tagger. This illustrates the situation that the relatively high complementary rate between tagger errors does not necessarily imply that there is anything to be gained by tagger plurality voting. To take advantage of the high complementary rates, it is necessary to employ a context-based combination, i.e to locate the contexts more "suitable" for the Exponential tagger and the contexts more "suitable" for the Markov model taggers.

Still, *How to improve the given results further?* Given the partial success of the plurality voting procedure, we applied it (and its variants) to combine Markov model taggers trained on partially different data produced by the so-called bagging procedure. The results have demonstrated that the Markov model methodology is not so heavily sensitive to the character of training data as we would wish[5]. More significantly, we have not noticed improvement of the tagging accuracy.

Since we wished to apply our results on languages other than Czech, we have adhered only to language-independent methods throughout this thesis. However, as the results show only small or no improvement, we believe that it is time to include language-dependent characteristics.

The next attempt to improve tagging accuracy should be driven by a context-dependent combination of taggers. Then, we should move from the language-independent ideas to the language-dependent ideas: we propose to start with the modification of the Czech positional tag set (we have been inspired here by the linguistic analysis in [Sgall, 1959]). All experiments

[5] I.e., the error complementary rates were relatively low (about 18%)

performed confirmed the highest error rate on the morphological categories `case`, `gender` and `number`; thus, we will separately merge those values of morphological categories `case`, `gender`, `number` that lead to an ambiguity into the one single `case` or `gender` or `number` value. Table 9.1 demonstrates a possible merging of the ambiguous `case` values. In addition to the basic seven `case` values, we add six more `case` values *a* through *f* as a disjunction of basic values; for instance, the ambiguity of the word forms in the vocative and in the nominative will be expressed e.g. by the `case` value *a*. The third column of the table specifies "restrictions" on the values of the part of speech and other MCs which are covered by the given `case` ambiguity. In fact, we have already used this technique from the beginning e.g. for the `gender` of active past participle verb forms intuitively knowing that this category might cause trouble; having done an error analysis now, we can target the affected categories much more effectively.

Given the proposed tag set modifications, the number of all possible tags as a whole increases, but the average number of tags for a given word form decreases. The ambiguity connected with the new `case`, `gender` and `number` values will then have to be resolved on the higher language levels of language analysis (syntactic, semantic). In other words, we will resolve (hopefully with high accuracy) as much ambiguity as we can at the tagging level, but no more.

`case` Values	Description	Restrictions
a	5 or 1[6]	[APC]***[15]**********[7] NN*[DP][15]**********[8] NNN[SDP][15]**********[9]
b	1 or 4[10]	[ANC]*N[14]***********[11] [AN]*F[DP][14]**********[12] [AN]*IS[14]**********[13]
c	4 or 2[14]	[AN]*MS[24]****[15]
d	4 or 7[16]	A*FS[47]**********[17]
e	6 or 2[18]	A**P[26]**********[19] NNFD[26]**********[20]
f	6 or 3[21]	NN[FM]S[36]**********[22]

Table 9.1: The ambiguity of the particular `case` values

[6] vocative or nominative
[7] adjective, pronouns, numerals in nominative or vocative
[8] nouns, dual or plural, nominative or vocative
[9] nouns, neuter, singular or dual or plural, nominative or vocative
[10] nominative or accusative
[11] adjectives or nouns or numerals, neuter, nominative or accusative
[12] adjectives or nouns, femininum, dual or plural, nominative or accusative
[13] adjectives or nouns, masculine inanimate, singular, nominative or accusative
[14] accusative or genitive
[15] adjectives or nouns, feminine, singular, genitive or accusative
[16] accusative or instrumental
[17] adjectives, feminine, singular, accusative or instrumental
[18] locative or genitive
[19] adjectives, plural, genitive or locative
[20] nouns, feminine, dual, genitive or locative
[21] locative or dative
[21] nouns, feminine or masculine animate, singular, dative or locative

Bibliography

[Bauer, Kohavi, 1998] E. Bauer and R. Kohavi. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning* 1-38, 1998.

[Baum, 1972] L. Baum. An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process. *Inequalities*, 3:1-8, 1972.

[Berger et al., 1996] A. Berger, S. D. Pietra, V. D. Pietra. A Maximum Entropy Approach to Natural Language Processing. In *Computational Linguistics*, 22(1), pp. 37-71, 1996.

[Bémová et al., 1999] A. Bémová, J. Hajič, B. Hladká and J.Panevová. Morphological and Syntactic Tagging of the Prague Dependency Treebank. In *Proceedings of ATALA Workshop*, pp. 21-29, Paris, France, 1999.

[Blum, Rivest, 1988] Blum and Rivest. Training 3-Node Neural Network is NP-complete. In *Proceedings of the 1988 Workshop on Computational Learning Theory*, 9-18. San Francisco, 1988.

[Böhmová, Panevová, Sgall, 1999] A. Böhmová, J. Panevová and P. Sgall. Syntactic Tagging: Procedure for the Transition from the Analytic to the Tectogrammatical Tree Structures. In *Proceedings of the 2nd International Workshop Text, Speech and Dialogue'99*, eds.V. Matoušek, P. Mautner, J. Ocelíková, P. Sojka, pp. 34-38, Plzeň, Czech Republic, 1999.

[Böhmová, Hajičová, 1999] A. Böhmová and E. Hajičová. How Much of the Underlying Syntactic Structure Can Be Tagged Automatically. In *Proceedings of ATALA Workshop*, pp. 31-39, Paris, France, 1999.

[Breiman, 1996] L. Breiman. Bagging Predictors. *Machine Learning* 24(2): 123-140, 1996.

[Brill, 1993a] E. Brill. *A Corpus-Based Approach to Language Learning.* A dissertation in Department of Computer and Information Science, University of Pennsylvania, Philadelphia, USA, 1993.

[Brill, 1993b] E. Brill. Automatic Grammar Induction and Parsing Free Text: A Transformation-Based Approach. In *Proceedings of the 3rd International Workshop on Parsing Technologies*, Tilburg, Netherlands, 1993.

[Brill, 1998] E. Brill and J. Wu. Classifier Combination for Improved Lexical Disambiguation. In *Proceedings of the COLING-ACL Conference*, pp. 191-195, Montreal, Canada, 1998.

[Chanod, Tapanainen, 1994] J.-P. Chanod and P. Tapanainen. Statistical and Constraint-Based Taggers for French. *Technical report MLTT-016*, Rank Xerox Research Centre, Grenoble, France, 1994.

[Collins Cobuild English Dictionary, 1995] *Collins Cobuild English Dictionary*, J. Sinclair(ed.) London: Harper Collins Publishers, Great Britain, 1995.

[Cutting at al., 1992] D. Cutting, J. Kupiec, J. Pedersen and P. Sibun. A Practical Part-of-Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy, 1992.

[Church, 1992] K. W. Church. Current Practice in Part of Speech Tagging and Suggestions for the Future. In Simmons(eds), *Studies in Slavic Philology and Computational Linguistics: In Honour of Henry Kučera*, pp. 13-48, Michigan Slavic Publications, 1992.

[Cuřín, Čmejrek, 1999] Automatic Translation Lexicon Extraction from Czech-English Parallel Texts. In *The Prague Bulletin of Mathematical Linguistics, 71*, pp. 47-57, Prague, Czech Republic, 1999.

[Daelemans, Zavrel, 1996] W. Daelemans and J. Zavrel. MBT: A Memory-Based Part of Speech Tagger-Generator. In *Proceedings of the Workshop on Very Large Corpora*, pp. 14-27, Copenhagen, Denmark, 1996.

[Dietterich, 1997] T. Dietterich. Machine-Learning Research: Four Current Directions. In *AI Magazine*, pp. 97-136, winter 1997.

[Dietterich, 1998] T. Dietterich. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning* 1-22, 1998.

[Elworthy, 1995] D. Elworthy. Tagset Design and Inflected Languages. In *Proceedings of the ACL SIGDAT Workshop, From Texts to Tags: Issues in Multilingual Language Analysis*, pp. 1-9, Dublin, Ireland, 1995.

[Forney, 1973] G. D. Forney, Jr. The Viterbi Algorithm. In *Proceedings IEEE, vol. 61*, pp. 268-278, 1973.

[Freund, Schapire, 1999] Y. Freund and R.E. Schapire. Experiments with a New Boosting Algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 148-156, 1999.

[Hajič, 1984] J. Hajič. *KODAS - A Simple Method of Natural Language Interface to a Database*. Explizite Beschreibung der Sprache und automatische Textbearbeitung XI. Faculty of Mathematics and Physisc, Charles University, Prague, Czech Republic, 1984.

[Hajič, 1994] J. Hajič. *Unification Morphology Grammar*. PhD thesis MFF UK, Prague, Czech Republic, 1994.

[Hajič, 1998] J. Hajič. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová*, ed. Eva Hajičová, pp. 106-132, Karolinum, Charles University Press, Prague, Czech Republic, 1998.

[Hajič, in press] J. Hajič. *Disambiguation of Rich Inflection (Computational Morphology of Czech).* Charles University Press - Karolinum, in press.

[Hajič, Hajičová, Panevová, Sgall, 1998] J. Hajič, E. Hajičová, J. Panevová and P. Sgall. Syntax v Českém národním korpusu [Syntax in the Czech National Corpus]. In *Slovo a slovesnost*, 3, LIX, pp. 168-177, 1998.

[Hajič, Hladká, 1997a] J. Hajič and B. Hladká. Probabilistic and Rule-Based Tagger of an Inflective Language - a Comparison. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pp. 111-118, Washington, USA, 1997.

[Hajič, Hladká, 1997b] Jan Hajič and Barbora Hladká. Morfologické značkování korpusu českých textů stochastickou metodou. In *Slovo a Slovesnost* 58, 4, pp. 288-304, AV ČR, Prague, Czech Republic, 1997.

[Hajič, Hladká, 1998a] J. Hajič and B. Hladká. Czech Language Processing - PoS Tagging. In *Proceedings of the First International Conference on Language Resources & Evaluation*, pp. 931-936, Granada, Spain, 1998.

[Hajič, Hladká, 1998b] J. Hajič and B. Hladká. Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In *Proceedings of COLING-ACL Conference*, pp. 483-490, Montreal, Canada, 1998.

[Hajič, Ribarov, 1997] J. Hajič and K. Ribarov. Rule-Based Dependencies. In *Proceedings of the Workshop on the Empirical Learning of Natural Language Processing Tasks*, pp. 125-136, Prague, Czech Republic, 1997.

[Hajič et al., 1998] J. Hajič, E. Brill, M. Collins, B. Hladká, D. Jones, C. Kuo, L. Ramshaw, O. Schwartz, C. Tillmann and D. Zeman. Core Natural Language Processing Technology Applicable to Multiple Languages: Workshop98 Final Report for the 1998 Language Engineering Workshop for Students and Professionals: Integrating Research and Education, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, Research Note 37, 1998.

[Hajičová et al., 1995] E. Hajičová, J. Borota, J. Hajič, M. Hnátková, V. Kuboň, K. Oliva, J. Panevová and P. Sgall. *Text-and-Inference Based Approach to Question Answering.* Theoretical and Computational Linguistics, volume 3, Prague, Czech Republic, 1995.

[Hajičová, 1998] E. Hajičová. Prague Dependency Treebank: From Analytic to Tectogrammatical Annotation. In *Proceedings of the First Workshop on Text, Speech, Dialogue*, pp. 45-50, Brno, Czech Republic, 1998.

[Hajičová, Panevová, Sgall, 1998] E. Hajičová, J. Panevová and P. Sgall. Language Resources Need Annotations to Make Them Really Reusable: The Prague Dependency Treebank. In *Proceedings of the First International Conference on Language Resources*, pp. 713-718, Granada, Spain, 1998.

[Halteren et al., 1998] H. van Halteren, W. Daelemans and J. Zavrel. Improving Data Driven Wordclass Tagging by System Combination. In *Proceedings of the COLING-ACL Conference*, pp. 491-497, Montreal, Canada, 1998.

[Henderson, Brill, 1999] J.C. Henderson and E. Brill. Exploiting Diversity in Natural Language Processing: Combining Parsers. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 187-194, University of Maryland, College Park, MD, USA, 1999.

[Hladká, 1994] B. Hladká. *Programové vybavení pro zpracování velkých českých textových korpusů* [Software Tools for Large Czech Corpora Annotation], MSc thesis MFF UK, Prague, Czech Republic, 1994.

[Hladká, 1999] B. Hladká. A Tagger Combination: A Method How to Get Better Results for an Inflective Languages. Final Report. 1999.

[Hladká, Ribarov,1998] B. Hladká and K. Ribarov. PoS Tags for Automatic Tagging and Syntactic Structures. In *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová*, ed. Eva Hajičová, pp. 226-240, Karolinum, Charles University Press, Prague, Czech Republic, 1998.

[Hyafil, Rivest, 1976] L. Hyafil and R. L. Rivest. Constructing Optimal Binary Decision Trees is NP-complete. *Information Processing Letters* 5(1):15-17, 1976.

[Jelinek, 1997] F. Jelinek. *Information Extraction from Speech and Text.* MIT Press, 1997.

[Jelinek, Mercer, 1980] F. Jelinek and R. L. Mercer. Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Proceeding of the Workshop on Pattern Recognition in Practice*, pp. 381-397, Amsterdam, Netherland, 1980.

[Kirschner, 1983] Z. Kirschner. *MOSAIC - A Method of Automatic Extraction of Significant Terms from Texts.* Explizite Beschreibung der Sprache und automatische Textbearbeitung X. Faculty of Mathematics and Physisc, Charles University, Prague, Czech Republic, 1983.

[Králíková, Panevová, 1990] K. Králíková and J. Panevová. *ASIMUT - A Method for Automatic Information Retrieval from Full Texts.* Explizite Beschreibung der Sprache und automatische Textbearbeitung XVII. Faculty of Mathematics and Physisc, Charles University, Prague, Czech Republic, 1990.

[Kuboň, Holan, Plátek, 1997] V. Kuboň, T. Holan and M. Plátek. A Grammar-Checker for Czech. *ÚFAL Technical Report, TR-1997-02*, Institute of Formal and Applied Linguistics, Charles University, Prague, Czech Republic, 1997.

[Maclin, Opitz, 1997] R. Maclin and D. Opitz. An Empirical Evaluation of Bagging and Boosting. In *Proceedings of the 14th National Conference on Artificial Intelligence*, pp. 546-551, Providence, Rhode Island, AAAI Press, 1997.

[Marcus, Santorini, Marcinkiewicz, 1993] M. M. Marcus, B. Santorini and M.-A. Marcinkiewicz. Building A Large Annotated Corpus of English. The Penn Treebank. *Computational Linguistics*, 20(2), pp. 313-330, 1993.

[Megyesi, 1999] B. Megyesi. Improving Brill's POS Tagger for an Agglutinative Language. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 275-284, University of Maryland, USA, 1999.

[Merialdo, 1994] B. Merialdo. Tagging English Text with a Probabilistic Model. In *Computational Linguistics*, 20(2), pp. 155-171, 1994.

[Mírovský, 1998] J. Mírovský. *Morfologické značkování textu: automatická disambiguace* [Morphological annotation of text: automatic disambiguation], MSc thesis MFF UK, Prague, Czech Republic, 1998.

[Quinlan, 1996] J. R. Quinlan. Bagging, Boosting, and c4.5. In *Proceedings of the Thirteenth National Conference on Artificial Conference*, pp. 725-730. AAAI/MIT Press.

[Ratnaparkhi, 1996] A. Ratnaparkhi. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of the First Empirical Methods in Natural Language Processing Conference*, pp. 133-141, Philadelphia, USA, 1996.

[Ribarov, 1996] K. Ribarov. *Automatická tvorba gramatiky přirozeného jazyka* [Automatic Natural Language Grammar Generation]. MSc. thesis MFF UK, Prague, Czech Republic, 1996.

[Sgall, 1959] P. Sgall. Soustava pádových koncovek v češtině. In *Acta Universitatis Carolinae - Philologica 2, Slavica Pragensia II.*, pp. 65-84, 1959.

[Schiller, 1996] A. Schiller. Multilingual Finite-State Noun Phrase Extraction. In *Proceedings of the ECAI'96*, Budapest, Hungary, 1996.

[Schmid, 1994] H. Schmid. Part-Of-Speech Tagging with Neural Networks. In *Procedings of the 15th COLING Conference*, pp. 172-176, Kyoto, Japan, 1994.

[Tapanainen, 1995] P. Tapanainen. RXRC Finite-State Compiler. *Technical Report MLTT-20*, Rank Xerox Research Center, Grenoble, France, 1995.

[Zeman, 1998] D. Zeman. A Statistical Approach to Parsing of Czech. In *The Prague Bulletin of Mathematical Linguistics, 69*, pp. 29-38, Prague, Czech Republic, 1998.

Appendix A

Penn Treebank Tag Set

1.	CC	Coordinating conjunction	25.	TO	“to’
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential “there”	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund or present participle
6.	IN	Preposition or subordinating conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd person singular present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd person singular present
9.	JJS	Adjective, superlative	33.	WDT	Wh-determiner
10.	LS	List item marker	34.	WP	Wh- pronoun
11.	MD	Modal	35.	WP$	Possessive wh-pronoun
12.	NN	Noun, singular or mass	36.	WRB	Wh-adverb
13.	NNS	Noun, plural	37.	#	Pound symbol
14.	NP	Proper noun, singular	38.	$	Dollar symbol
15.	NPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	”	Straight double quote
21.	RBR	Adverb, comparative	45.	‘	Left open single quote
22.	RBS	Adverb, superlative	46.	“	Left open double quote
23.	RP	Particle	47.	’	Right close single quote
24.	SYM	Symbol	48.	”	Right close double quote

Appendix B

Subtag Level Errors Produced by MM Taggers on Particular Part of Speech

Tagger	POS	SubPOS	g	n	c	possg	possn	d	a	s
T0	16	16	258	182	434	0	0	16	2	0
T1	18	18	280	196	468	0	0	18	3	0
T2	15	15	297	186	446	0	0	15	1	0
T3	17	17	283	204	477	0	0	17	1	0
T4	18	18	282	202	478	0	0	18	2	0
T5	17	17	282	194	452	0	0	17	1	0
T6	21	21	285	181	458	0	0	21	2	0
T7	19	19	276	205	468	0	0	19	2	0
T8	20	20	289	190	477	0	0	20	1	0
T9	18	18	288	196	486	0	0	18	2	0
T10	20	20	273	187	460	0	0	20	3	0
T11	21	21	276	175	454	0	0	20	2	0
T12	19	19	281	197	465	0	0	19	2	0
T13	18	18	279	180	456	0	0	18	3	0
T14	17	17	292	194	468	0	0	17	1	0
T15	20	20	282	188	469	0	0	20	1	0
T16	22	22	292	185	468	0	0	22	1	0
T17	19	19	279	196	458	0	0	19	1	0
T18	18	18	287	181	469	0	0	18	1	0
T19	21	21	286	196	446	0	0	21	1	0
T20	21	21	274	187	441	0	0	21	1	0
Total number of adjectives in test data: 3,914										

Table B.1: MM taggers: errors on adjectives

POS	SubPOS	g	n	c	s
3	3	21	7	45	0
3	3	22	7	56	0
4	4	23	6	46	0
4	4	17	8	48	0
2	2	21	6	52	0
4	4	20	6	38	0
0	0	19	7	44	0
2	2	22	7	46	0
1	1	16	7	44	0
3	3	23	7	49	0
1	1	19	6	51	0
3	3	24	8	48	0
2	2	21	7	50	0
1	1	17	7	52	0
2	2	26	7	51	0
3	3	21	7	52	0
3	3	20	5	53	0
2	2	23	6	47	0
3	3	16	8	39	0
4	4	22	8	52	0
2	2	23	7	46	0
Total number of numerals in test data: 1,439					

Table B.2: MM taggers: errors on numerals

POS	SubPOS	d	a	s
74	97	16	9	1
77	102	18	11	1
73	97	18	12	1
72	98	19	12	1
71	94	16	9	1
76	99	17	10	1
71	97	15	8	1
73	98	14	7	1
70	91	14	7	1
72	94	15	8	1
71	94	15	8	1
74	98	16	9	1
76	99	17	10	1
76	101	17	10	1
73	97	17	10	1
72	96	15	8	1
75	10	19	12	1
77	101	17	10	1
77	104	18	11	2
72	95	16	9	1
74	97	15	8	1
Total number of adverbs in test data: 1,818				

Table B.3: MM taggers: errors on adverbs

POS	SubPOS
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
Total number of interjections in test data: 7	

Table B.4: MM taggers: errors on interjections

POS	SubPOS
49	49
51	51
51	51
49	49
50	50
49	49
48	48
50	50
51	51
50	50
53	53
51	51
50	50
47	47
54	54
48	48
48	48
53	53
51	51
54	54
53	53
Total number of conjunctions in test data: 1,847	

Table B.5: MM taggers: errors on conjunctions

POS	SubPOS	g	n	c	a	s
122	122	248	390	1120	84	1
125	125	252	416	1206	88	2
123	123	252	411	1179	85	1
119	119	256	406	1189	84	1
124	124	246	403	1181	84	1
131	131	255	410	1163	91	1
124	124	252	408	1175	87	1
123	123	240	389	1184	85	1
134	134	264	426	1209	91	1
123	123	245	422	1215	87	1
128	128	244	400	1185	90	1
133	133	263	407	1204	93	1
122	122	242	405	1196	84	1
121	121	256	405	1184	83	1
123	123	255	394	1212	88	1
125	125	249	400	1167	86	1
122	122	247	392	1161	85	1
127	127	251	410	1185	91	1
121	121	249	407	1205	82	2
129	129	255	409	1185	91	2
120	120	239	402	1154	84	1
Total number of nouns in test data: 10,203						

Table B.6: MM taggers: errors on nouns

POS	SubPOS	g	n	c	possg	possn	p
86	93	102	131	224	1	1	1
87	94	119	141	242	1	1	1
90	100	114	148	252	3	3	3
87	95	118	138	252	2	2	2
89	100	110	144	245	2	2	2
88	97	117	149	241	1	1	1
86	92	114	147	254	1	1	1
92	101	112	145	244	2	2	2
86	97	112	154	245	1	1	1
85	96	109	142	239	2	2	2
89	98	115	141	250	4	4	4
88	100	109	143	246	4	4	4
93	101	104	144	242	2	2	2
85	95	105	136	233	2	2	2
88	97	104	133	243	2	2	2
91	97	108	139	242	0	0	0
90	99	110	142	252	1	1	1
88	98	107	147	240	3	3	3
89	97	105	133	231	2	2	2
89	96	118	143	258	2	2	2
86	96	103	136	235	3	3	3
Total number of pronouns in test data: 2,123							

Table B.7: MM taggers: errors on pronouns

POS	SubPOS	c
7	7	147
7	7	169
11	11	163
11	11	169
7	7	166
7	7	155
7	7	152
7	7	168
8	8	158
10	10	170
10	10	163
8	8	156
9	9	160
9	9	155
9	9	171
9	9	163
11	11	161
7	7	151
9	9	167
8	8	165
9	9	160
Total number of prepositions in test data: 3,455		

Table B.8: MM taggers: errors on prepositions

POS	SubPOS
9	9
9	9
7	7
9	9
9	9
8	8
9	9
7	7
8	8
8	8
9	9
8	8
9	9
8	8
9	9
10	10
7	7
7	7
9	9
9	9
9	9
Total number of particles in test data: 143	

Table B.9: MM taggers: errors on particles

POS	SubPOS	g	n	p	t	a	v
67	67	5	89	47	61	40	61
77	77	7	96	56	68	44	68
73	73	5	92	49	66	43	66
72	72	7	98	51	64	44	64
71	71	5	89	48	65	43	65
69	69	5	89	51	63	37	63
70	70	5	89	47	65	43	65
79	79	7	96	55	70	46	70
78	78	7	86	54	72	47	72
73	73	5	87	53	66	41	66
65	65	5	88	46	59	37	59
76	76	6	98	52	67	44	67
72	72	6	89	49	62	44	62
68	68	7	90	47	64	44	64
73	73	7	96	51	66	43	66
70	70	6	87	50	68	42	68
77	77	6	93	51	70	49	70
75	75	6	97	51	68	45	68
76	76	5	92	54	68	46	68
77	77	6	86	54	70	43	70
79	79	8	95	54	71	47	71
Total number of verbs in test data: 4,278							

Table B.10: MM taggers: errors on verbs

POS	SubPOS
6	6
6	6
6	6
6	6
6	6
5	5
6	6
4	4
6	6
5	5
5	5
6	6
6	6
6	6
5	5
5	5
4	4
6	6
6	6
4	4
5	5
Total number of unknowns in test data: 745	

Table B.11: MM taggers: errors on unknowns

Appendix C

Case, Number and Gender Complementary Rates

	$T0_{MM}$	$T1_{MM}$	$T2_{MM}$	$T3_{MM}$	$T4_{MM}$	$T5_{MM}$	$T6_{MM}$	$T7_{MM}$	$T8_{MM}$	$T9_{MM}$
$T0_{MM}$	0.00	77.23	75.68	79.07	62.79	81.59	62.35	66.76	69.33	75.24
$T1_{MM}$	79.10	0.00	78.03	86.62	83.10	91.77	86.84	86.26	83.41	84.17
$T2_{MM}$	77.03	77.39	0.00	86.09	79.68	88.38	81.56	86.64	84.71	91.90
$T3_{MM}$	80.68	86.54	86.40	0.00	85.82	86.40	79.83	83.23	83.23	90.47
$T4_{MM}$	65.36	82.86	79.97	85.70	0.00	88.68	71.18	74.02	69.10	82.27
$T5_{MM}$	82.29	91.38	88.16	85.83	88.30	0.00	83.27	89.28	87.47	85.79
$T6_{MM}$	64.29	86.40	81.48	79.27	70.63	83.50	0.00	75.97	69.21	82.67
$T7_{MM}$	68.97	86.02	86.79	83.04	73.95	89.60	76.35	0.00	73.09	81.95
$T8_{MM}$	71.65	83.30	85.04	83.21	69.32	87.95	70.00	73.35	0.00	86.61
$T9_{MM}$	77.34	84.21	92.15	90.55	82.57	86.47	83.28	82.31	86.74	0.00
$T10_{MM}$	71.97	82.05	88.91	84.23	80.33	89.82	73.60	75.10	84.60	77.15
$T11_{MM}$	60.16	78.23	83.51	81.07	66.67	86.86	73.08	73.17	69.56	82.70
$T12_{MM}$	77.28	75.84	76.74	80.67	81.48	88.35	76.65	82.07	81.39	91.37
$T13_{MM}$	81.09	73.00	77.41	89.60	85.56	91.21	85.42	88.59	83.85	85.60
$T14_{MM}$	73.33	80.09	87.69	87.38	80.93	91.82	78.98	81.29	77.51	80.40
$T15_{MM}$	72.01	90.96	85.21	86.39	74.61	80.41	72.24	80.00	77.81	79.50
$T16_{MM}$	85.07	81.34	83.57	85.89	86.80	86.21	85.62	86.66	85.39	86.57
$T17_{MM}$	64.82	80.38	79.73	82.48	75.66	87.42	69.53	73.19	75.43	83.49
$T18_{MM}$	78.70	80.60	72.79	84.16	84.93	86.37	80.10	81.05	83.66	90.48
$T19_{MM}$	76.15	83.76	91.58	91.00	78.28	87.96	81.86	80.23	82.71	71.36
$T20_{MM}$	60.99	83.06	78.11	81.15	74.29	86.93	73.26	72.47	73.36	87.68

Table C.1: MM Taggers: case complementary rate: part I

	$T10_{MM}$	$T11_{MM}$	$T12_{MM}$	$T13_{MM}$	$T14_{MM}$	$T15_{MM}$
$T0_{MM}$	70.11	57.27	75.63	80.09	70.93	70.30
$T1_{MM}$	82.44	78.57	76.21	73.90	80.08	91.20
$T2_{MM}$	88.83	83.30	76.43	77.53	87.32	85.17
$T3_{MM}$	84.48	81.26	80.86	89.89	87.30	86.67
$T4_{MM}$	80.47	66.71	81.51	85.84	80.65	74.92
$T5_{MM}$	89.56	86.44	87.98	91.10	91.43	80.01
$T6_{MM}$	73.30	72.61	76.24	85.43	78.26	72.06
$T7_{MM}$	75.21	73.13	82.04	88.78	80.96	80.19
$T8_{MM}$	84.82	69.82	81.55	84.28	77.34	78.24
$T9_{MM}$	77.69	83.02	91.53	86.12	80.44	80.09
$T10_{MM}$	0.00	81.96	84.73	85.10	75.56	83.05
$T11_{MM}$	82.07	0.00	78.14	83.15	77.10	77.46
$T12_{MM}$	84.82	78.14	0.00	79.58	86.36	84.64
$T13_{MM}$	84.91	82.84	79.21	0.00	84.68	91.86
$T14_{MM}$	76.09	77.47	86.58	85.20	0.00	83.47
$T15_{MM}$	82.97	77.21	84.47	91.92	83.01	0.00
$T16_{MM}$	91.35	84.84	77.20	81.70	88.35	81.98
$T17_{MM}$	78.50	71.59	81.43	87.60	71.36	74.89
$T18_{MM}$	86.06	84.34	73.06	81.45	87.86	85.88
$T19_{MM}$	74.66	81.72	90.09	88.19	76.24	74.75
$T20_{MM}$	79.65	66.03	81.71	86.89	75.64	77.37

Table C.2: MM Taggers: case complementary rate: part II

	$T16_{MM}$	$T17_{MM}$	$T18_{MM}$	$T19_{MM}$	$T20_{MM}$
$T0_{MM}$	84.11	62.74	77.13	74.47	59.50
$T1_{MM}$	81.77	80.92	80.88	84.04	83.86
$T2_{MM}$	83.48	79.73	72.40	91.49	78.54
$T3_{MM}$	86.14	82.87	84.30	91.10	81.93
$T4_{MM}$	86.92	76.00	84.93	78.35	75.15
$T5_{MM}$	85.88	87.19	85.93	87.60	86.95
$T6_{MM}$	85.48	69.39	79.73	81.57	73.67
$T7_{MM}$	86.75	73.50	81.00	80.24	73.32
$T8_{MM}$	85.62	75.95	83.79	82.89	74.43
$T9_{MM}$	86.92	83.99	90.64	71.93	88.29
$T10_{MM}$	91.37	78.65	85.96	74.56	80.19
$T11_{MM}$	84.96	71.95	84.33	81.75	67.12
$T12_{MM}$	77.37	81.66	73.04	90.11	82.29
$T13_{MM}$	81.51	87.53	81.09	87.99	87.07
$T14_{MM}$	88.62	72.18	88.04	76.67	76.80
$T15_{MM}$	81.92	74.93	85.71	74.52	77.85
$T16_{MM}$	0.00	84.43	76.88	83.52	83.93
$T17_{MM}$	84.35	0.00	83.62	83.94	71.73
$T18_{MM}$	77.08	83.84	0.00	90.48	84.30
$T19_{MM}$	83.62	84.12	90.45	0.00	86.20
$T20_{MM}$	83.53	71.16	83.76	85.77	0.00

Table C.3: MM Taggers: case complementary rate: part III

	$T0_{MM}$	$T1_{MM}$	$T2_{MM}$	$T3_{MM}$	$T4_{MM}$	$T5_{MM}$	$T6_{MM}$	$T7_{MM}$	$T8_{MM}$	$T9_{MM}$
$T0_{MM}$	0.00	42.54	43.83	58.23	45.50	46.66	37.40	51.93	52.96	54.76
$T1_{MM}$	46.72	0.00	55.42	55.66	55.66	64.96	51.37	61.38	58.28	69.01
$T2_{MM}$	47.29	54.89	0.00	69.36	59.47	61.88	54.52	61.52	60.55	64.41
$T3_{MM}$	61.36	55.77	69.80	0.00	64.68	72.53	57.67	67.54	69.20	76.46
$T4_{MM}$	48.42	54.74	59.12	63.87	0.00	65.69	54.01	55.23	55.11	71.29
$T5_{MM}$	49.94	64.54	61.88	72.14	65.98	0.00	56.69	66.34	71.17	53.92
$T6_{MM}$	40.10	49.82	53.63	56.21	53.51	55.84	0.00	58.92	53.26	63.22
$T7_{MM}$	54.45	60.54	61.14	66.75	55.18	66.02	59.32	0.00	63.70	64.43
$T8_{MM}$	56.53	58.43	61.16	69.24	56.18	71.62	54.87	64.61	0.00	75.06
$T9_{MM}$	57.89	68.90	64.71	76.32	71.77	54.31	64.23	65.07	74.88	0.00
$T10_{MM}$	43.46	52.80	48.69	63.39	58.41	58.78	47.07	61.64	63.64	61.39
$T11_{MM}$	51.97	52.46	59.71	61.18	49.88	63.76	55.90	53.44	59.09	62.90
$T12_{MM}$	57.56	58.65	67.35	52.48	62.39	71.10	61.06	63.85	65.54	71.46
$T13_{MM}$	45.29	55.58	57.47	63.61	53.20	61.48	52.45	60.23	57.34	59.22
$T14_{MM}$	45.85	55.14	58.74	63.82	53.16	61.21	54.15	59.48	59.73	67.41
$T15_{MM}$	56.11	63.84	67.21	72.44	58.98	53.37	59.35	65.34	67.08	50.75
$T16_{MM}$	63.08	62.70	59.82	71.71	70.84	55.19	58.57	68.71	68.59	51.94
$T17_{MM}$	44.29	54.29	55.24	64.40	54.29	60.83	52.50	60.48	60.48	68.81
$T18_{MM}$	41.51	50.19	55.02	61.83	48.33	57.37	50.56	55.89	62.45	66.17
$T19_{MM}$	72.78	78.01	80.32	71.45	75.82	66.10	74.73	82.26	82.75	67.56
$T20_{MM}$	42.63	51.05	53.41	64.06	57.37	60.72	51.80	57.37	60.97	60.10

Table C.4: MM Taggers: number complementary rate: part I

	$T10_{MM}$	$T11_{MM}$	$T12_{MM}$	$T13_{MM}$	$T14_{MM}$	$T15_{MM}$
$T0_{MM}$	41.65	49.74	54.88	43.96	43.83	54.76
$T1_{MM}$	54.83	53.87	59.24	57.81	56.85	65.44
$T2_{MM}$	50.30	60.43	67.43	59.11	59.83	68.28
$T3_{MM}$	65.04	62.43	53.27	65.52	65.28	73.72
$T4_{MM}$	59.37	50.36	62.17	54.62	54.01	59.98
$T5_{MM}$	60.07	64.41	71.17	62.97	62.24	54.89
$T6_{MM}$	47.72	55.84	60.39	53.38	54.49	59.90
$T7_{MM}$	62.48	53.84	63.58	61.39	60.17	66.14
$T8_{MM}$	65.32	60.45	66.15	59.62	61.40	68.65
$T9_{MM}$	62.92	63.88	71.77	61.12	68.54	52.75
$T10_{MM}$	0.00	61.64	60.52	55.42	54.67	65.50
$T11_{MM}$	62.16	0.00	59.71	50.12	50.61	65.48
$T12_{MM}$	61.67	60.34	0.00	62.27	65.42	73.88
$T13_{MM}$	55.08	49.06	60.85	0.00	51.82	64.74
$T14_{MM}$	54.89	50.19	64.56	52.42	0.00	64.19
$T15_{MM}$	65.46	64.96	73.07	64.96	63.97	0.00
$T16_{MM}$	59.70	63.70	67.83	63.58	64.83	54.82
$T17_{MM}$	59.05	53.93	61.19	57.98	47.02	64.29
$T18_{MM}$	50.68	57.62	59.98	55.51	51.92	63.57
$T19_{MM}$	78.25	77.40	66.83	75.21	76.55	66.71
$T20_{MM}$	52.04	53.66	59.98	54.40	50.06	68.28

Table C.5: MM Taggers: number complementary rate: part II

	T16$_{MM}$	T17$_{MM}$	T18$_{MM}$	T19$_{MM}$	T20$_{MM}$
T0$_{MM}$	62.08	39.85	39.33	71.21	40.49
T1$_{MM}$	64.48	54.23	52.09	78.43	52.92
T2$_{MM}$	61.28	54.64	56.21	80.46	54.64
T3$_{MM}$	73.13	64.45	63.38	72.06	65.52
T4$_{MM}$	71.65	53.28	49.27	75.79	58.15
T5$_{MM}$	56.82	60.31	58.50	66.34	61.76
T6$_{MM}$	59.29	50.92	50.92	74.42	52.15
T7$_{MM}$	69.55	59.56	56.64	82.22	58.10
T8$_{MM}$	70.19	60.57	64.01	83.14	62.59
T9$_{MM}$	54.07	68.66	67.34	68.06	61.48
T10$_{MM}$	59.90	57.16	50.44	77.71	51.81
T11$_{MM}$	64.37	52.46	57.99	77.15	54.05
T12$_{MM}$	68.92	60.58	60.94	66.99	60.94
T13$_{MM}$	63.49	55.71	54.96	74.40	53.83
T14$_{MM}$	65.18	44.86	51.92	76.08	50.06
T15$_{MM}$	54.99	62.59	63.34	65.84	68.08
T16$_{MM}$	0.00	67.46	65.46	66.83	61.83
T17$_{MM}$	69.05	0.00	49.17	77.02	52.86
T18$_{MM}$	65.80	47.09	0.00	75.84	49.07
T19$_{MM}$	67.80	76.55	76.31	0.00	77.28
T20$_{MM}$	62.21	50.93	49.07	76.83	0.00

Table C.6: MM Taggers: number complementary rate: part III

	$T0_{MM}$	$T1_{MM}$	$T2_{MM}$	$T3_{MM}$	$T4_{MM}$	$T5_{MM}$	$T6_{MM}$	$T7_{MM}$	$T8_{MM}$	$T9_{MM}$
$T0_{MM}$	0.00	65.03	60.41	77.69	62.18	78.64	55.92	75.24	67.76	62.86
$T1_{MM}$	67.47	0.00	74.30	84.43	73.67	89.49	79.87	83.54	72.78	75.06
$T2_{MM}$	63.58	74.59	0.00	83.10	67.33	89.61	72.59	86.11	70.21	81.73
$T3_{MM}$	79.11	84.33	82.80	0.00	80.00	88.15	74.65	88.66	81.02	89.55
$T4_{MM}$	63.80	72.92	66.02	79.56	0.00	88.28	73.18	83.07	59.38	83.07
$T5_{MM}$	79.90	89.37	89.37	88.09	88.48	0.00	84.38	96.54	84.64	79.26
$T6_{MM}$	58.19	79.48	71.74	74.32	73.42	84.26	0.00	80.77	68.39	75.74
$T7_{MM}$	76.15	82.96	85.45	88.34	82.96	96.46	80.47	0.00	81.39	84.14
$T8_{MM}$	70.04	72.82	69.91	81.16	60.56	84.83	69.03	82.05	0.00	81.54
$T9_{MM}$	64.86	74.65	81.21	89.45	83.27	79.15	75.80	84.43	81.21	0.00
$T10_{MM}$	75.40	84.13	80.03	97.35	88.49	94.58	81.61	67.46	92.06	75.66
$T11_{MM}$	63.83	69.04	66.62	77.03	63.71	85.15	71.07	77.41	59.01	81.98
$T12_{MM}$	59.74	67.89	72.37	75.39	70.66	87.37	62.37	81.71	73.16	76.45
$T13_{MM}$	68.20	67.67	70.17	84.49	70.96	83.44	73.72	90.80	67.94	69.91
$T14_{MM}$	82.06	79.77	90.33	95.80	92.49	96.56	80.41	72.39	88.17	82.19
$T15_{MM}$	66.93	81.19	75.62	82.88	74.58	82.36	59.14	81.19	72.37	72.63
$T16_{MM}$	83.18	85.75	82.54	90.89	82.03	95.89	76.77	77.15	78.56	89.73
$T17_{MM}$	60.65	74.71	68.26	80.90	78.32	87.35	61.03	76.00	73.68	76.77
$T18_{MM}$	64.85	77.56	70.17	73.28	67.70	87.29	68.22	79.51	77.82	73.80
$T19_{MM}$	84.34	85.34	83.83	96.24	79.70	92.98	85.84	76.32	82.71	82.58
$T20_{MM}$	79.87	91.26	82.65	89.54	85.17	92.72	78.15	72.05	81.72	85.30

Table C.7: MM Taggers: gender complementary rate: part I

	$T10_{MM}$	$T11_{MM}$	$T12_{MM}$	$T13_{MM}$	$T14_{MM}$	$T15_{MM}$
$T0_{MM}$	74.69	61.22	58.37	67.07	80.82	65.31
$T1_{MM}$	84.81	69.11	69.11	68.86	79.87	81.65
$T2_{MM}$	81.10	67.08	73.72	71.59	90.49	76.47
$T3_{MM}$	97.45	76.94	76.18	84.97	95.80	83.18
$T4_{MM}$	88.67	62.76	70.96	71.22	92.32	74.48
$T5_{MM}$	94.75	85.02	87.71	83.87	96.54	82.59
$T6_{MM}$	82.06	70.58	63.10	74.19	80.13	59.35
$T7_{MM}$	67.76	76.67	81.78	90.83	71.56	81.00
$T8_{MM}$	92.41	59.17	74.21	69.15	88.24	73.07
$T9_{MM}$	76.32	81.72	76.96	70.53	81.98	72.84
$T10_{MM}$	0.00	91.40	86.24	85.32	72.88	85.98
$T11_{MM}$	91.75	0.00	67.77	70.56	87.94	75.76
$T12_{MM}$	86.32	66.58	0.00	74.61	85.39	70.00
$T13_{MM}$	85.41	69.51	74.64	0.00	81.34	80.29
$T14_{MM}$	73.92	87.91	85.88	81.93	0.00	85.11
$T15_{MM}$	86.25	75.23	70.43	80.54	84.82	0.00
$T16_{MM}$	83.70	78.43	78.43	85.37	79.33	75.87
$T17_{MM}$	82.71	65.55	69.68	73.55	72.52	69.42
$T18_{MM}$	80.42	71.98	71.21	77.17	83.66	76.91
$T19_{MM}$	80.83	82.21	86.97	81.70	77.69	87.22
$T20_{MM}$	80.66	77.35	82.78	85.43	74.30	80.79

Table C.8: MM Taggers: gender complementary rate: part II

	$T16_{MM}$	$T17_{MM}$	$T18_{MM}$	$T19_{MM}$	$T20_{MM}$
$T0_{MM}$	82.18	58.50	63.13	82.99	79.32
$T1_{MM}$	85.95	75.19	78.10	85.19	91.65
$T2_{MM}$	82.98	69.21	71.21	83.85	83.60
$T3_{MM}$	90.96	81.15	73.76	96.18	89.94
$T4_{MM}$	81.77	78.12	67.58	78.91	85.42
$T5_{MM}$	95.90	87.45	87.45	92.83	92.96
$T6_{MM}$	76.65	61.03	68.39	85.42	78.71
$T7_{MM}$	76.67	75.62	79.29	75.23	72.35
$T8_{MM}$	78.89	74.21	78.38	82.55	82.55
$T9_{MM}$	89.70	76.83	74.00	82.11	85.71
$T10_{MM}$	83.20	82.28	80.03	79.76	80.69
$T11_{MM}$	78.68	66.12	72.59	81.98	78.30
$T12_{MM}$	77.89	69.08	70.79	86.32	82.89
$T13_{MM}$	85.02	73.06	76.87	80.81	85.55
$T14_{MM}$	79.52	72.90	83.97	77.35	75.32
$T15_{MM}$	75.62	69.26	76.91	86.77	81.19
$T16_{MM}$	0.00	78.69	83.83	76.64	78.95
$T17_{MM}$	78.58	0.00	69.55	89.03	78.32
$T18_{MM}$	83.66	69.39	0.00	83.27	80.29
$T19_{MM}$	77.19	89.35	83.83	0.00	79.07
$T20_{MM}$	78.28	77.75	79.87	77.88	0.00

Table C.9: MM Taggers: gender complementary rate: part III