Creation of Lexicons and Language Models for Automatic Broadcast News Transcription

Ph.D. Thesis

Author:	Ing. Dana Nejedlová
Doctoral degree program:	P2612 Electrical Engineering and Informatics
Specialization:	2612V045 Technical Cybernetics
State doctoral exam	Liberec, March 10, 2004
Work station:	Department of Informatics Faculty of Economics Technical University of Liberec
Supervisor:	Prof. Ing. Jan Nouza, CSc.

Size of Thesis and Its Appendixes

Number of pages:	88
Number of pictures:	13
Number of tables:	26
Number of formulas:	49
Number of appendixes:	1

+ AUTOREFERÁT

KES

88 s.
obr. tab.

Declaration

I hereby declare that this thesis is the result of my own original work, and where it draws on the work of others, this is acknowledged at the appropriate points in the text.

Liberec, January 30, 2006

Ing. Dana Nejedlová

Dana Nejedlová

Acknowledgements

I would like to thank the following people for enabling me to start and complete this thesis:

Doc. Ing. Jaromír Kukal, Ph.D. from the Institute of Chemical Technology in Prague for introducing me into the field of Artificial Intelligence,

Doc. Ing.Vladimír Kracík, CSc. from the Technical University of Liberec for his lessons in Mathematics and for recommending me to Professor Nouza,

Prof. Ing. Jan Nouza, CSc. for his scientific guidance,

all members of SpeechLab for their collegiality, namely to Ing. Petr Červa for providing me with the recordings of radio broadcast news and his software for preparation of speech databases, and to Ing. Jan Kolorenč for providing me with the outcomes of his system for phonetic transcription and for running the recognizer for me,

Ing. Robert Vích, DrSc. from the Institute of Radio Engineering and Electronics of the Academy of Sciences of the Czech Republic for his kind interest in my scientific growth through the years,

Mr. Richard Lombardi for careful proofreading of this thesis,

and last but not least to my mother, father, and grandmother for their love and both material and moral support.

Anotace

Tvorba slovníků a jazykových modelů pro automatický přepis zpravodajských pořadů

Disertační práce

Ing. Dana Nejedlová

Tato disertační práce se zabývá jazykovou částí problému automatického rozpoznávání zpravodajských pořadů.

Kapitola č. 1 obsahuje motivaci k řešení problému rozpoznávání souvislé řeči.

Kapitola č. 2 se snaží vysvětlit složitost úlohy rozpoznávání souvislé řeči.

Kapitola č.3 představuje systém pro automatický přepis zpravodajských pořadů vyvinutý v laboratoři SpeechLab.

Kapitola č. 4 podává přehled publikací z jiných laboratoří týkajících se tématu této disertační práce.

Kapitola č. 5 uvádí cíle této disertační práce. Tyto cíle jsou:

1. Příprava textového korpusu,
2. Sestavení slovníku obsahujícího několik stovek tisíců slov,
3. Fonetická transkripce slov ve slovníku,
4. Výpočet různých bigramových jazykových modelů,
5. Příprava testovací databáze promluv,
6. Testování slovníku, jazykových modelů a parametrů rozpoznávače na testovací databázi,
7. Vytvoření kritérií pro měření kvality výstupu rozpoznávače.

Kapitola č. 6 vysvětluje teorii nezbytnou k dosažení cílů této práce a popisuje aplikaci této teorie na konkrétní problémy řešené v laboratoři SpeechLab. Tato kapitola je strukturována podle cílů disertační práce.

Kapitola č.7, která je také strukturována podle cílů této práce, uvádí výsledky experimentů.

Kapitola č. 8 shrnuje, co bylo uděláno během výzkumu popsaného v této práci a co by mělo být uděláno v budoucnu.

Annotation

Creation of Lexicons and Language Models for Automatic Broadcast News Transcription

Dissertation Thesis

Ing. Dana Nejedlová

This dissertation thesis deals with the linguistic part of the problem of automatic recognition of broadcast news.

Chapter 1 contains the motivation to solve the problem of continuous speech recognition.

Chapter 2 tries to explain the complexity of the problem of continuous speech recognition.

Chapter 3 introduces the system for broadcast news transcription developed at SpeechLab.

Chapter 4 reviews various publications from other laboratories concerning the topic of this thesis.

Chapter 5 presents goals of this thesis. The goals are:

1. Preparation of text corpus,
2. Compilation of vocabulary containing several hundred thousand words,
3. Phonetic transcription of the words in the vocabulary,
4. Computation of various bigram language models,
5. Preparation of test speech database,
6. Testing of vocabulary, language model, and recognizer's parameters on the test speech database,
7. Developing the criteria for measuring the quality of the recognizer's output.

Chapter 6 explains theory necessary to reach the goals of this thesis and describes the application of this theory on particular problems solved in SpeechLab. This chapter is structured according to the thesis goals.

Chapter 7, which is also structured according to the goals of this thesis, presents experimental results.

Chapter 8 summarizes what has been done in the research described in this thesis and what should be done in the future.

List of Abbreviations

ARPA	Advanced Research Projects Agency
ASR	automatic speech recognition
BNT	broadcast news transcription
CPU	central processing unit (processor)
CSR	continuous speech recognition
DARPA	The Defense Advanced Research Projects Agency
HMM	hidden Markov model
LDC	Linguistic Data Consortium
LM	language model
LVCSR	large vocabulary continuous speech recognition
MFCC	mel frequency cepstral coefficients
MLE	maximum likelihood estimate
OOV	out of vocabulary (missing in the vocabulary)
PAC	Phonetic Alphabet for Czech
PC	personal computer
PMI	Pointwise Mutual Information
SRI	Stanford Research Institute
WER	word error rate

Contents

List of Figures

List of Tables

Chapter 1
Introduction

This thesis deals with automatic writing down of spoken news from audio signal. The field of speech processing has a common term for it, “broadcast news transcription”. It is a special case of the more common task called “continuous speech recognition”.

Continuous speech recognition (CSR) is as well a special case of speech recognition. Automatic speech recognition (ASR) began in the 1950s when Bell Labs introduced the first speech recognition system that was able to discern digits (numbers) spoken by a single speaker, with long isolated (discrete) pauses [1]. Not less than 20 years had elapsed until the first systems capable of continuous speech recognition appeared. This thesis somewhat enlightens the reason why this development has taken such a long time.

The first widely known project of continuous speech recognition was started in the USA in 1971. It was called Speech Understanding Research (SUR) project and was funded by the U.S. Department of Defense’s Advanced Research Projects Agency (ARPA—later renamed DARPA). The results of this project were systems called HEARSAY-II and HARPY, both publicized in 1980. See [1], [2], and [3] for details.

Broadcast news transcription is a part of activities that become more important with the increasing amount of multimedia (audio & video) information that is generated by our civilization. We should be able to retrieve useful information from our vast resources of archived news in audio & video format. Broadcast news transcription turns the audio part of the data into text. There is also the possibility of using the video part of the signal, see e.g. [4], in solving this task. Once the information is in text form, it can be prepared for information retrieval using key words. Perhaps, the most important project of this kind is called “Informedia” (Integrated Speech, Image and Language Understanding for Creation and Exploration of Digital Video Libraries) established by Carnegie Mellon University and launched in 1994 [5].

This thesis deals with broadcast news transcription that utilizes only audio data. Considering the many subtasks involved, like signal preprocessing, training of acoustic models, or recognizer designing, we focus here on lexical, phonetic, and grammatical aspects of the problem.

We should also explain the terms “recognizer” and “speech recognition system”. These terms are synonyms, and we understand by them software that is run on a personal computer. For the sake of completeness, we should also mention that whenever we use the term “recognition”, we mean by it “automatic recognition”, i.e. recognition performed by some automatic recognizer. Speech recognition/recognizer is sometimes also called “speech decoding/decoder”. Recognition is mapping of the acoustic speech signal to text without the need to understand the meaning.

Chapter 2
Why Is Continuous Speech Recognition So Difficult?

2.1. Discrete Speech Recognition

The first recognizers were able to recognize words spoken with pauses. Uttering a single word and waiting for the recognizer's response or speaking with regular pauses is called "discrete speech". Sound is the changing of air pressure in time. Computer representation of sound is a succession of numbers expressing the values of air pressure measured in discrete regular intervals dense enough to capture the meaning or reproducibility of sound. We call this representation "acoustic signal". Silence (caused by making pauses between words) is easy to recognize. That is how we can automatically divide the signal into segments that are supposed to mean single words.

The recognizer has a list of words into which it can classify the input words. This list is called the lexicon, vocabulary or dictionary. Each word in the list has an acoustic model that resides in the recognizer's memory. The signal segment representing a single word is compared with every acoustic model. The word that belongs to the most similar model is chosen for the output. We must take into account that the same word can be spoken in a number of fashions even if it is uttered by the same person. That is why discrete recognizers have often many models representing the same word.

To make the process of matching of models with acoustic signals more feasible, both the signal and the models are represented in a compressed form that is composed of the succession of the so-called "acoustic feature vectors". These feature vectors and the methods of their computation are a result of laborious research that had in scope finding out how to extract features that are crucial for the description of acoustic form of speech.

Now we describe briefly the advances that had to be achieved to accomplish discrete speech recognition and in the next section we compare them with the findings necessary for continuous speech recognition.

The list of advances necessary for discrete speech recognition:

1. Computer representation of sound
2. Feature extraction
3. Algorithms for matching succession of features with models of words

2.2. Continuous Speech Recognition

Continuous speech is natural speech. In normal speech in most languages words follow each other without interruptions. People can discern word boundaries only when they know the vocabulary and sometimes also the meaning. (In some languages they can also orient according to stresses and melody.)

The initial research of continuous speech recognition, having in its foundation the findings of discrete speech recognition, had to consider that continuous speech recognition is a task where the right succession of the right words should be found that minimizes the distance between the acoustic model of this succession and the succession of features of input acoustic signal. It should also be considered that single words in the recognizer's vocabulary usually have many acoustic models. Such a task can be solved by search methods known from the scientific field of artificial intelligence. The aim of these methods is usually to solve combinatorial problems that are impossible to solve by brute force algorithms that would find optimal solutions. Instead, they try to find sub-optimal (close to the best solution) solutions by reasonably quick algorithms. They offer the usual tradeoff between the time spent on computing and precision.

The input signal must be divided into sequences containing several words. A sentence may be an ideal unit. Exceedingly long sequences may take too much time to process. Broadcast news transcription (BNT) as a sub-field of continuous speech recognition requires also the exclusion of the audio segments that do not contain speech because the recognizer could decipher these segments into words (even if the common practice for the recognizers is that they have also acoustic models of noises). Omitting of non-speech segments is important also for saving the time for speech recognition because many applications require real-time processing.

Initial recognition experiments using all the processes mentioned above had very poor results, because the search space was too large. Such problems can be solved by changing the representation of data rather than by finding new search algorithms. The new representation of acoustic models called Hidden Markov Models (HMM) has significantly improved the recognition. According to [2] and [3], HMMs were first applied to speech recognition in the 1970s.

HMMs decompose words into states changing in time. These states can be determined by the so-called vector quantization but they may also correspond to phonemes. Phonemes are sound units of speech defined by the scientific field of phonetics. The speech-recognizing community has finally adopted the practice of dividing the speech signal into phonemes, quantizing feature vectors of single phonemes, assigning each phoneme to its HMM that has a small number of states (usually 3), and training HMMs of individual phonemes on a large number of examples of these phonemes in various contexts and sound conditions to make them robust.

Acoustic models of words are now a concatenation of HMMs representing the phonemes. Words in recognizer's vocabulary are linked to their acoustic models via the so-called phonetic transcriptions. Phonetic transcription is a succession of phonemes that constitute the acoustic form of the word. Phonetic transcription is also the process of assigning the succession of phonemes to words. This process should be automated because the number of words in the vocabulary may be very large. Details about automatic phonetic transcription are a part of this thesis.

Many words are phonetically similar. Humans understand speech by matching the hypotheses of perceived words with usual context in which they have heard these words before. The human mind has an incredible capacity for processing this kind of information. This ability may be the reason, why computers still lag behind humans in speech recognition. It has become inevitable to incorporate some knowledge about usual context of words into speech recognizers. This knowledge is called the language model (LM) and can be in the form of either templates of parts of speech that describe possible grammatical structures, or probabilities of the succession of selected words. The former type is called a rule-based LM. The latter type of LM, called *n*-gram LM, is more flexible and more widely used. The *n*-gram language model was introduced into speech processing in the 1950s but its advantage over some other ways of representation of grammatical knowledge was not fully recognized until the 1970s [6] (pp. 230 – 231). Selected kinds of *n*-gram language models are a subject of detailed investigation in the following parts of this thesis.

With solving more demanding tasks, like recognition of speech on unpredictable topics, the methods of building of recognizer's vocabularies has gained importance. Today, words for the lexicons for recognizers are selected from the most frequent words found in very large amounts of electronic text called "text corpus" or simply "corpus". The corpus is also used for the assessment of probabilities in the *n*-gram language models mentioned above. In the time when the field of speech recognition had just begun, only a limited quantity of texts were available in electronic form. Hence, another reason why successful realization of continuous speech recognition has taken such a long time. Building of the vocabulary is also a subject of this thesis.

Finally, a sizeable amount of programming and recognition experiments should be done to make all these new discoveries work well together. The basic principles of either discrete and continuous speech recognition are compiled for Czech readers in [7].

To sum it up, continuous speech recognition needs the following findings and works:

1. Computer representation of sound
2. Segmentation of a continuous input signal into several word long segments
3. Feature extraction
4. Hidden Markov Models representation of words and algorithms for working with HMMs
5. Segmentation of an acoustic signal of words into phonemes to obtain data for HMMs training
6. Training of HMMs representing phonemes
7. Phonetic transcription
8. Language modeling
9. Corpus processing
10. Lexicon building
11. Recognizer building and tuning

At minimum, items 7 and 8 represent processes that are very language dependent. That is why findings in speech recognition of some language are not fully applicable to some other languages. The greatest progress has been done in the recognition of speech in English. The reason for this is not only that English is the native language of many workers of the best speech processing laboratories funded by the most developed industrial countries, but also because English has some features that make its recognition easier than some other languages. Methods, like *n*-gram language models, work well for English but are less effective for languages similar to Czech. A good compilation of reasons for this is given in [8].

Chapter 3
System for Broadcast News Transcription Developed at SpeechLab

3.1. History

Research described in this thesis has been done at the Czech speech-processing laboratory called SpeechLab. It was founded by professor Jan Nouza in 1994, and works on the ground of the Faculty of Mechatronics of the Technical University of Liberec. The main research domain of this laboratory is speech recognition of Czech language, which is not only theoretical but has also some practical applications, see [9].

The research in SpeechLab started with the recognition of isolated words and short phrases. The first system for continuous speech recognition was developed there in 1999. It was tested on the recognition of speech about time and date information. It had a rule-based language model, which meant that it could handle only a limited set of possible grammatical structures. With the lexicon of about 200 words it could work in real time with an accuracy of 90% correctly recognized words. [7]

In 2001 the system evolved into a recognizer that could work with lexicons containing several thousand words. Acoustic models of words for this system were concatenations of context-free phoneme models trained on 20 hours of speech by 80 different speakers. The language model of this system was based on bigrams (a kind of *n*-gram LM) estimated on a 55-milion-word corpus of newspapers and novels. The test database of this system contained 800 sentences read from newspapers by 20 non-professional speakers with no background noise. These 800 sentences were composed of 3,622 different words. These words formed the vocabulary of the recognizer. This means that there were no out-of-vocabulary words in this task, so the experiments with this system and the test database did not simulate real-life conditions of recognition of unpredictable speech. The purpose of these experiments was to find an optimal set of various parameters of the recognizer. The best results of recognition were about 64% of correctly recognized words as reported in [10] and [11]. The results in more detail can be seen in Chapter 7.4.1.

In 2003 SpeechLab developed a continuous speech recognizer that could work with lexicons containing several tens of thousands of words. Experiments with this system finally simulate real life because its vocabulary was independent of the recognized texts. Dictation of 100 newspaper sentences (each sentence was dictated as a whole into this system) resulted in an accuracy of 70% of correctly recognized words with a 30,000-word vocabulary that did not cover 14% of words in the recognized text. The system could work in real time. [12]

Information about the next stage of the system was published in 2004 in [13] and [14]. This time the test data were recordings of three complete real Czech broadcast news TV shows from three different Czech stations (1 hour in total, 8,451 words). Their preprocessing was fully automated. Only the opening and closing jingles were removed, the rest was automatically segmented according to the speaker changes. The resulting longest utterance had as much as 181 words. The recognition had an accuracy of 71% of correctly recognized words, with a 200,000-word vocabulary that did not cover less than 2% of words in the recognized text. The transcription was available in approximately 6 times longer than the duration of the recognized utterances. Compared to the system from 2003 mentioned in the previous paragraph, the improvement of accuracy was poor. The reason for this was the quality of the recognized recordings. Only 44% of them constituted clear speech read by professional speakers in the studio. Recognition of those parts had an accuracy of 82%.

Currently SpeechLab recognizes continuous speech with a 312,000-word vocabulary. The vocabulary covers more than 99% of the test data mentioned in the previous paragraph, and the accuracy of recognition of the whole test database (the previously mentioned three Czech broadcast news TV shows) is 78%. [15]

3.2. Principles

All speech-recognizing systems developed in SpeechLab have always been a result of its members' programming efforts. Many other laboratories use commercially available or free software. Such approaches may be advantageous at the beginning of the research but in the end they may be restrictive. Commercial or free software may not support as large vocabularies as those needed by inflectional languages like Czech or may not be so flexible as to allow experimenting with some non-standard language models, or such software may be simply less effective in terms of utilizing available hardware.

Principles of SpeechLab's own speech decoder were first published in [16] and the details about its tuning in [17]. The description given below is detailed enough to illustrate problems of lexicon and language model building, which is the subject of this thesis. Figure 1 shows the scheme of the decoder.

The input signal is sampled at 16 bit / 8 kHz rate and pre-processed to get 39-feature MFCC frame vectors. All the words in the recognizer's vocabulary are represented by concatenated phoneme models. The phoneme models correspond to the 3-state Hidden Markov Models trained on a database of 45 hours of annotated speech that is a mix of microphone and broadcast signal. The HMMs are 32-mixture monophones.

Having an utterance represented by a time sequence of feature vectors $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T$, we are searching for an unknown word sequence that maximizes probability

$$P(w_1^*, w_2^*, ..., w_N^*) = \underset{w,N}{Max} \prod_{n=1}^{N} g(w_{n-1}, w_n) \cdot V(w_n) \qquad (1)$$

The first term $g(w_{n-1}, w_n)$ in equation (1) represents the language model, i.e. bigram probability that word w_n follows w_{n-1}. The second term $V(w_n)$ is equal to the acoustic score of word w_n achieved by its HMM evaluated on a certain part of vector sequence $\mathbf{X}$. To prevent floating-point operation underflow caused by the successive multiplication of small values, the formula (1) should be converted to a logarithmic scale. Because of the quite different natures of the language model $g(w_{n-1}, w_n)$ and the acoustic score $V(w_n)$, the term for the language model gets a weight called λ or *LM Factor*. According to some of our own and the other author's observations, the recognition system has a tendency of preferring shorter words to longer ones. To suppress this phenomenon we have introduced the parameter called *Word Insertion Penalty* or *WP* that worsens each candidate word's score. The result is equation (2):

$$P(w_1^*,...,w_N^*) = \underset{w,N}{Max} \sum_{n=1}^{N} (\lambda \cdot \ln g(w_{n-1}, w_n) + WP + \ln V(w_n)) \qquad (2)$$

Equation (2) is solved by applying the Viterbi algorithm, described in [7] and [6] (pp. 176 – 180) augmented by HMM pruning and hypotheses pruning techniques.

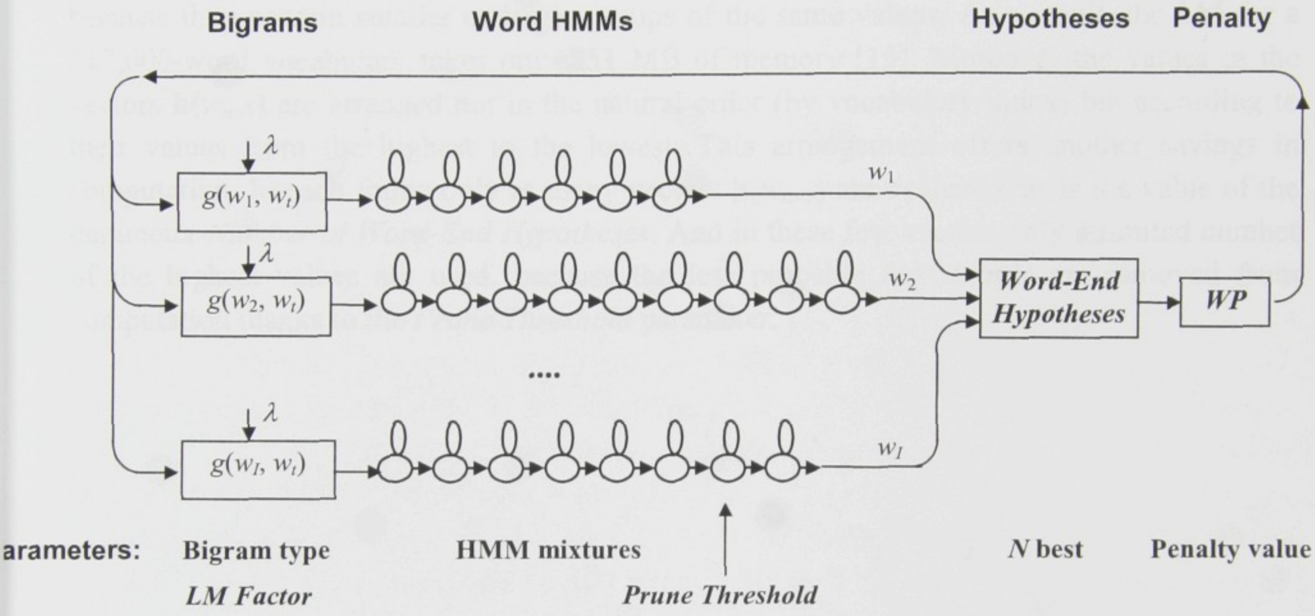


Figure 1. The structure of the word sequence recognition procedure with all key parameters [11]

Acoustic models of all words are evaluated in parallel with the assumption that an arbitrary word can begin at an arbitrary frame. The frame is a 10-ms-long segment of the acoustic input signal. Each frame is converted into an acoustic feature vector in the course of the process called parameterization. In each frame less probable threads of the Viterbi algorithm are cut off to speed up the computation. This is controlled by the parameter called *Prune Threshold*. The words whose score computed by the Viterbi algorithm divided by the best existing score is smaller than the *Prune Threshold* are temporarily removed from computation. The highest scores in the ending states of the HMMs belong to the most probable words that end in the given frame. Only a limited number of the best word-candidates are selected for further computation. This number of words is determined by the parameter called *Number of Word-End Hypotheses*. Each candidate's score is penalized by the constant called *WP* or *Word Insertion Penalty*. In the next computation the hypotheses about all possible next words are taken into account. The initial score of a new word is equal to the previously ended word's score to which the bigram probability $\lambda \cdot \ln g(w_{n-1}, w_n)$ of the new word on condition that the previously ended word precedes it is added.

The part of the decoder described above, that is the most related to the subject of this thesis, is the bigram language model. When the working vocabulary of the decoder has 312,000 words, the bigram language model is a table of probabilities of succession of every possible word-pair of this vocabulary, so this table has the second power of 312,000 values. When each value is represented by 4 bytes in the computer, the whole model would occupy 363 GB of the computer's memory. Due to the hardware constraints of current PC technology, it is necessary to compress such a table.

The description of this compression is given in [16]. The compression takes advantage of the fact that many values in the smoothed language model are the same. The table of the bigram model is divided into vectors $\mathbf{h}(w_{n-1})$ of the values of conditional probabilities of word-pairs that share the same previous word. The vectors can be efficiently compressed because they contain smaller or larger groups of the same values. As a result, the LM for a 312,000-word vocabulary takes only 251 MB of memory [15]. Moreover the values in the vectors $\mathbf{h}(w_{n-1})$ are arranged not in the natural order (by vocabulary index) but according to their values from the highest to the lowest. This arrangement offers another savings in computation. In each frame only as many vectors $\mathbf{h}(w_{n-1})$ are evaluated as is the value of the parameter *Number of Word-End Hypotheses*. And in these few vectors only a limited number of the highest values are used, because the less probable next words are removed from computation thanks to the *Prune Threshold* parameter.

Chapter 4
The Latest Advances of Broadcast News Transcription

Proceedings of two important regular conferences Eurospeech/Interspeech and Text, Speech and Dialogue (TSD) have been studied to get informed about what the other laboratories are doing in the field which is the subject of this thesis. The main contribution that can be obtained from this study is the following: There are two main approaches to BNT which is very close to the subject called "large vocabulary continuous speech recognition" (LVCSR) that is also worthy of being given attention. The first approach is to work with commercially available or freely available software. This approach is mentioned also in Chapter 3.2. And the second approach is to develop the laboratory's own speech recognizers.

Let us first have a look at the first approach. This thesis somewhat follows the results of the Ph.D. thesis [8]. The work [8] describes LVCSR with the use of the commercially available AT&T decoder published in [18]. This decoder is built on the basis of the finite-state machines framework, and, at least in the time when the work [8] was being written (around 2002 –2003), it could work with vocabularies containing at most 60 thousand items. A vocabulary of 60 thousand wordforms is not enough for acceptably high coverage of Czech texts. The thesis [8] (p. 66) states that a vocabulary of this size with the most frequent Czech words covers about 92% of words in the independent Czech text while the same-sized vocabulary with English words covers 99% of English text. This lower coverage is caused by the inflected character of the Czech language which means that from a single Czech lexical lemma (a basic wordform) many different wordforms can be derived. To overcome the problem with the low coverage, Czech words were divided into their stems and endings (suffixes). A vocabulary of 60 thousand most frequent stems and endings had according to [8] (p. 83) 96% coverage. From the vocabulary of 60 thousand words a trigram language model was compiled. Trigram is a conditional probability of a certain word on condition that a certain succession of two words precedes it. The results of [8] show that the accuracy of recognition is higher by 2% (73% is the absolute value.) when using the trigram language model computed from the 60-thousand-word vocabulary of stems and endings in comparison to the same LM computed from the vocabulary of 60 thousand most frequent Czech wordforms. The accuracy of 73% of correctly recognized words was the best result that was achieved in [8]. The work [8] in its conclusion on page 86 also mentions that even four-gram LM was tested, but it has not brought any noticeable gain in the recognition accuracy. The study of work [8] suggests that any further improvement of accuracy with the use of the software and vocabulary reported in [8] may not be possible. On the other hand, BNT of English with a vocabulary of 65 thousand words is performed with the accuracy of 80 to 86% [19].

One of the most important projects of LVCSR of Czech language is MALACH. Its purpose is to provide access to the collection of 116,000 hours of video with 52,000 interviews ("testimonies") in 32 languages of personal memories of survivors of the World War II Holocaust. The facts about the origin of the video collection can be learned at www.vhf.org. The homepage of the project MALACH is www.clsp.jhu.edu/research/malach/. The goal of MALACH is to help with cataloguing of the video by automatic speech recognition (ASR). Many research teams are partaking in this project trying to perform ASR in many languages. Czech researchers working with Czech-spoken part of the collection are among them. Article [20] describes linguistic problems of this task, some of them being similar to the BNT task, for example the existence of many unknown personal and geographical names and foreign words. ASR in [20] is performed with a 61-thousand-word vocabulary. The main contribution of this article from the language-modeling point of view is the method of extraction of relevant parts of the available text corpus. The problem is that the text corpus used for language modeling consists mainly of news, but the subject of speech recognition are personal stories in spontaneous, spoken Czech. Using the vocabulary of the transcribed training part of the testimonies, relevant (in-domain) sentences from the corpus were extracted. Addition of these sentences to the transcriptions of the training part of the testimonies from which the language model was trained has increased the accuracy of recognition by 2%, which is a significant improvement.

The language that is very close by its nature to Czech and that has recently been a subject of LVCSR experiments is the Slovenian language. Articles [21], [22], and [23] show that the Slovenian researchers have worked with various publicly available recognizers with a vocabulary of 20 thousand items. To get a higher coverage of the test speech database, the words in the vocabulary are decomposed to their stems and endings. The accuracy of recognition is about 45% and the time consumed by the decoding is relatively high.

Another inflectional language is Greek. Articles [24] and [25] present the following facts about LVCSR in Greek. The speech recognition engine used was the SRI Decipher [26] working with a 60-thousand-word lexicon. A lexicon of that size covers 96.5% of newspaper text. It is a higher coverage than the 92%-coverage of the Czech lexicon of 60 thousand wordforms mentioned above in this chapter, which was also measured on newspaper text. This can mean that Czech language is more inflectional than Greek, but it should be taken in mind that coverage is highly sensitive on the topic and style of the text. Four kinds of trigram back-off language models based on wordforms were created. Each kind of LM was smoothed by a different smoothing method. The smoothing methods used ordered from the best to the worst are: Modified Kneser-Ney (18.57), Good-Turing (19.59), Witten-Bell (19.84), and Absolute Discounting (20.78). The numbers in parentheses are word error rates (WER). WER is equal to 100 – accuracy, i.e. the percentage of misrecognized words in the text. All these smoothing methods are described in [8] except of the Modified Kneser-Ney which can be studied from [27]. The WER measure was then lowered by 0.28% by applying the maximum entropy smoothing that combines the original Modified Kneser-Ney LM with a class-based LM. The classes were the words with the same stem. Thesis [8] describes maximum entropy models on pages 32 and 33 and mentions that they are very computationally intensive.

Another group of languages that are hard to process by ASR due to the high number of wordforms are agglutinative (sometimes called compounding) languages. According to [28] Turkish, Hungarian, Finnish, Estonian, Dutch, German, Japanese, and Korean rank among them. The common task that must be solved for these languages is their decomposition into word-components to get a reasonable coverage of independent texts, and subsequent compounding of the recognizer's output into compound words. Research in ASR of these languages is characterized by many original approaches and several cases when a laboratory has developed its own speech recognizer. This brings us to the description of the second approach.

ASR of Turkish is published in [29], [30]. These articles describe original decoder based on the finite state networks and word-splitting methods.

ASR of Hungarian is published in [31]. The authors have developed their own decoder based on the architecture of the weighted finite-state transducer.

An original way of building topic-based LMs using a neural computing method called "Self-Organizing Maps" for Finnish language is published in [32]. The Finnish recognizer that can produce an unlimited vocabulary is published in [33].

ASR of Estonian is published in [34]. The authors used the Japanese recognizer Julius described in [35].

ASR of Dutch is performed by the Dutch recognizer developed by the ESAT-PSI speech group. Articles [36] and [37] concentrate mainly on compounding and decompounding methods. Article [37] recommends that highly frequent compound words should not be split in the vocabulary and the LM. The authors also plan to take the opposite approach that consists in the combination of frequent orthographic word tuples, referred to as multi-words, collocations, or frequent word sequences, into single items in the recognition lexicon as it is recommended in [38].

German language has severe problems with coverage of independent texts by a given lexicon of wordforms. Articles [25] and [36] present comparative tables of English, French, Greek, Dutch, and German languages showing that with 60 thousand words only 95.1% of words in German newspapers can be covered. German language is according to these data the worst of these languages in terms of coverage. While the previously mentioned compounding languages were recognized with vocabularies containing at most 60 thousand word-components, the recognition of German must use larger vocabularies even if their items are also formed of the parts of split words. Article [39] describes the specifics of German language and shows that recognition with the vocabulary of 150 word-components has a better accuracy than recognition with the vocabulary of 200 thousand wordforms including compounds. The recognition engine used was based on BBN's Rough'n'Ready suite of technologies (the Byblos BNT system) [40]. Article [41] presents the BNT system for German language working with a vocabulary of 300 thousand wordforms. This vocabulary does not contain decomposed words. The recognizer used is a product of the French laboratory LIMSI. Article [42] presents a decompounding algorithm for German compound words. This algorithm was used to assist the development of pronunciation dictionaries of the 300-thousand vocabulary.

BNT of Japanese with the purpose to provide TV news with closed caption in real time is presented in [43]. The Byblos English BNT system [40] is used for this task. It has a working dictionary of 62 thousand words that are the result of Japanese morphological analysis. The morphological analysis is necessary, because the Japanese text does not contain word boundaries. Article [43] states that it has been empirically found that the WER must be less than 5% and the average word latency has to be less than 2 seconds to allow sufficiently quick manual error correction before the caption is presented. The authors are very close to meeting these two requirements. Their results are better than the results of English BNT which are usually presented as the DARPA Hub-4 Broadcast News Transcription task, see e.g. [19]. The explanation why the recognition of Japanese has better results is also given in [43]. Article [44] gives an interesting idea how to improve the accuracy of broadcast news recognition by employing a single person who listens to the news and re-speaks them to the recognizer that is trained to that person's voice. This idea is applicable to all the other languages as well. Re-speaking (or rephrasing) does not mean repeating. A skilled re-speaker knows the vocabulary of the recognizer and knows how to formulate the original sentences so that they become concise and fit well into the caption format, because providing the subtitles is again the target of this research. The Japanese concentrate on this task also because typing Japanese is much more time-consuming than typing English.

BNT of Korean is published in article [45]. The problem with Korean is the very short length of compounds in this compounding language. The article shows that the accuracy of recognition increases when these short morphemes are concatenated into longer vocabulary items according to statistical measures. The authors use their own recognizer presented in [46].

To complete this chapter, we should also look at the major languages not mentioned above. Article [47] shows methods of lexicon adaptation for the BNT task in Italian. For the experiments the ITC-irst Italian BNT system with a 64-thousand-word lexicon was used. The accuracy of recognition is about 25%.

Article [48] presents the research behind the implementation of closed-captioning in French. The Canadian team from the CRIM laboratory uses their own recognizer for this purpose. They have also adopted the re-speak method proposed in [44]. The size of the recognizer's vocabulary is only 20 thousand wordforms. It covers about 94% of text. The accuracy of recognition of this preliminary research is about 30%.

And finally, Spanish BNT is described in [49]. The authors use BBN's Rough'n'Ready suite of technologies with the Byblos BNT system [40] that has been used also for the recognition of German and Japanese as mentioned above. The recognition with a 73-thousand-word vocabulary had the accuracy of 16%. The authors have tried to decompose inflectional verbs that are in Spanish called "cliticized verbs". However, their experimental results of recognition with the LM made of the decomposed verbs are not encouraging.

Chapter 5
Thesis Goals

The research of current activities in BNT and LVCSR presented in Chapter 4 suggests that inflectional languages must be recognized with very large vocabularies that consist of several hundred thousand items. Morphological decomposition of these languages, especially Slavic languages like in this thesis studied Czech or the abovementioned Slovenian, results in a large amount of very short word-components in the resulting decomposed vocabulary. But many studies cited above have shown that short words are highly confusable in recognition. The other problem with such decomposition is the fact that the trigram language model that is usually used for decomposed lexicons cannot capture the succession of two consecutive wordforms. By the term "wordform" we always mean an undecomposed word i.e. whichever succession of letters that does not contain a space character that has appeared in the corpus. If the two consecutive wordforms were both decomposed in stem and ending, then the trigram LM would capture either the stem and ending of the first wordform and the stem of the second wordform or the ending of the first wordform and the stem and ending of the second wordform.

The research described in this thesis is based on the assumption that the words are natural linguistic units carrying semantic, syntactic, and grammatical information encoded into a string of phonetic events. All these four attributes (semantic, syntactic, morphological, and phonological) are closely interrelated and can be uniquely represented on the word level rather than on sub-word (morpheme) or super-word (class) levels [50]. To achieve good results by this method of representation the following tasks must be completed:

1. Preparation of text corpus
2. Compilation of vocabulary containing several hundred thousand words
3. Phonetic transcription of the words in the vocabulary
4. Computation of various bigram language models
5. Preparation of test speech database
6. Testing of vocabulary, language model, and recognizer's parameters on the speech database
7. Developing the criteria for measuring the quality of the recognizer's output

Chapter 6
Theory and Its Application

6.1. Text Corpus

6.1.1. The Purpose of Text Corpus

The purpose of text corpus in ASR is to get information about frequencies of single words and successions of a few words. The list of the most frequent words is a source of the recognizer's vocabulary. The list of word-successions and their frequencies is the resource for the computation of the *n*-gram language model. Text corpus must be in the form of electronic text so that it can be processed by the computer. The corpus should be stylistically close to the texts that will be automatically recognized and at the same time it should be very large. It is usually difficult to meet the both requirements. The texts that will be recognized are usually in the style of spoken language but most of the available corpus consists of written language.

Collecting of quality corpora for minor languages like Czech is a laborious process while major languages have already large annotated corpora available for the research community. Czech is a language with a very rich vocabulary in comparison to for example English. The richness of the Czech language is caused its inflective nature. The English language is also very rich, but this is caused by the fact that English is spoken by so many people in so many parts of the world who deal with so many things. However, the most frequent English words can cover more texts than Czech words as can be read in Chapter 4. Language models of lexically rich languages must be computed from relatively large corpora. This means that it is difficult for Czech automatic speech recognition to compete with major languages that have large corpora available and at the same time smaller vocabularies.

6.1.2. Corpus Cleaning

Czech electronic newspapers on the Internet are the most important resource of text corpus in our lab. The original format of the text is HTML, so the first operation that has to be done is the removal of HTML tags. [51] The resulting plain text must be cleaned. The process of corpus cleaning is the following [15]:

1. Each sentence is put on a separate line in the final corpus. The identification of sentences is automatic, and the algorithm that does this contains many rules telling what period really marks the end of a sentence.
2. Single words in brackets are then deleted. Such words are usually useless abbreviations.
3. Repeating headers, footers, and formatting characters are deleted.
4. Indeclinable abbreviations are expanded.
5. Expressions of the type *x-letý* (*x-year old*), where *x* is written in digits, are expanded.
6. Every word is converted to lower case, and every punctuation mark is surrounded by a space character to enable their counting and co-occurrence analysis. For the purpose of

training of the *n*-gram model and lexicon building all words in the corpus should be in lower case. See Chapter 6.1.5 for details.

7. Numbers meaning hours and dates, and some other numbers are rewritten to their spoken form. Ordinal numbers preceded by a preposition are expanded to their correct grammatical case and gender with the use of the Czech morphological analyzer [52]. The details are given below.
8. Words are rewritten to their standard orthographic form. Since many words, mainly those of foreign origin, may have alternative spelling, we have made the unification of the orthography towards the most frequent variants. This also helped to make the lexicons slightly smaller and more compact. [13] We have manually found 35,000 wordforms that should be rewritten. These rewriting rules are also applied to the reference transcriptions used for the evaluation of the recognition tests. [50] This procedure is usually called orthography normalization. Chapter 7.2.3 shows how this normalization increases the coverage of the text by the lexicon. Table 25 in Chapter 7.7 shows the improvement of the accuracy of recognition and the out-of-vocabulary (OOV) rate when the reference transcriptions are normalized.
9. Collocations, i.e. phrases of words that often appear together, are joined by a special character so that they are treated as a single word during the training of a language model. Currently we have 1,700 collocations in our lexicon. See Chapter 6.2.10 for information about the importance of collocations and about finding them. Chapter 7.4.2 shows the improvement of the accuracy of recognition thanks to joining the collocations into single words.

Point number 7 deserves a more detailed explanation. We rewrite ordinals written by digits to their spoken form, because their pronunciation is dependent on their spelling, and the information about pronunciation is necessary for the recognizer. In this way the corpus gets closer to the spoken language. The morphological analyzer is needed for this task, because Czech is an inflected language. The following two example sentences illustrate the cases when a preposition precedes an ordinal number:

Vlak přijede ***na 3.*** *nástupiště.* should be rewritten as *Vlak přijede* ***na třetí*** *nástupiště.*
(*The train will arrive at the third platform.*)
Martin si vsadil ***na 3.*** *koně.* should be rewritten as *Martin si vsadil* ***na třetího*** *koně.*
(*Martin has bet on the third horse.*)

The wordforms *třetí* and *třetího* in these example sentences are ordinal numerals meaning *the third*. The word *na* is a preposition. The wordforms *třetí* and *třetího* can be written in the form of digits followed by a period (e.g. *3.*) that indicates that they are ordinals. If a sentence contains such ordinals in the form of digits, they may have many different spellings. For example the ordinal *3.* may be rewritten into the following wordforms or inflections: *třetí*, *třetího*, *třetímu*, *třetím*, *třetích*, *třetími*, *třetíma*. The choice of the proper wordform of the ordinal depends on the preposition before the ordinal and the grammatical category of the noun that follows the ordinal. According to the database of the Czech morphological analyzer

[52] the Czech language has approximately 70 prepositions that can be followed by the wordforms in up to 3 different grammatical cases. The Czech morphology distinguishes between 7 grammatical cases (1. nominative, 2. genitive, 3. dative, 4. accusative, 5. vocative, 6. locative, 7. instrumental), two numbers (singular and plural), and three genders (masculine, feminine, neuter). The combination of case, number and gender determines the correct wordform of the ordinal. The case is determined by the preposition. The case, number and gender of the ordinal must be the same as the case, number and gender of the noun following the ordinal. This is an example of the so-called grammatical agreement in the Czech language that is mentioned also in connection with the language modeling in Chapter 6.4. We have compiled the list of all possible prepositions with all possible cases they relate to. The input of the morphological analyzer is the noun after the ordinal. The output of the morphological analyzer is the list of all possible combinations of case, number and gender of the input noun. From these combinations those that have the grammatical cases that belong to the preposition before the ordinal are selected. The ordinal that should be expanded is rewritten into the selected combinations of case, number, and gender. If all resulting wordforms of the ordinal are identical, then the ordinal is rewritten to that wordform, otherwise it is untouched. Rules for rewriting ordinals to their correct wordforms of a certain case, number and gender, have been compiled manually. In our example the preposition *na* can relate to either accusative or locative cases. The word *nástupiště* (*platform*) has either singular or plural number, neuter gender, and several cases (either nominative, genitive, accusative, or vocative). The ordinal *3.* in the accusative case and in both singular and plural number and neuter gender should be spelled as *třetí*. That is why it can be rewritten to that wordform. The second example sentence will not be rewritten, because the wordform *koně* (*horse*) can be either in singular number and the genitive or accusative case, or in plural number and the nominative or accusative or vocative case. The gender of the wordform *koně* is masculine. The ordinal *3.* in the accusative case and in singular number and masculine gender has the wordform *třetího*. But the same ordinal in the accusative case and in plural number and masculine gender has the wordform *třetí*. This is an example of ambiguity that would deserve a more complicated analysis that is not performed in our current state of research. The abovementioned procedure can be applied also to declinable abbreviations, e.g. *sv.* (*saint*) can be expanded to *svatý*, *svatého*, *svatému*, *svatém*, *svatým*, *svatá*, *svaté*, *svatou*, *svatí*, *svatých*, *svatými*, *svatýma*. In some sentences the ordinals and declinable abbreviations are not preceded by some preposition. Such sentences again would deserve a deeper semantic analysis.

Numbers that are supposed to be in the nominative grammatical case can also have more than one way of pronunciation, for example, *1625* can be pronounced as *šestnáct set dvacet pět* or *tisíc šest set dvacet pět*, *25* can be pronounced as *pětadvacet* or *dvacet pět*. [51] We transcribe them into the most frequent variants of pronunciation.

See Chapter 7.1.1 for the results of rewriting numbers in our corpus.

There are some other types of errors in the corpus as well. Article [53] tries to classify them. Typographical errors, especially hyphens between the syllables of a word when the word is divided at the end of a line, cause either a loss of affected words or their confusion with some correct word. The other types or errors mentioned in [53] are either unimportant for our purpose (punctuation errors, stylistic errors) or we have at best resigned ourselves to be unable to identify and correct them (morphological, syntactic, and semantic errors). [51]

6.1.3. The Size of Our Text Corpus

Our corpus has been built and regularly updated from almost all text sources available in electronic format [50]. Recently (in 2005) it contains about 2.6 GB of plain text data. After the process of cleaning described in Chapter 6.1.2 we have found in our corpus 360,104,333 words, from which 2,099,353 were distinct. The corpus consists of Czech newspapers (84%), transcripts of TV and radio news (9%), parliament speeches (4%), and electronic books (3%) that have been available on the Internet in the period of 1992 – 2004 [15]. The transcripts of TV and radio news and parliament speeches are the most valuable parts of our corpus because they are the closest to the task of BNT for which we use the corpus. The transcriptions of the recordings that we use for our recognition tests are not a part of them, because we obey the methodology of training sets and test sets described in the following Chapter 6.1.4.

6.1.4. The Methodology of Training Sets and Test Sets

The methodology of training sets and test sets must be observed in any research area dealing with training to obtain some parameters that should be finally used in classification of in advance unknown data. Its application to speech processing that classifies speech audio signal into words is described in [6] (p. 204) and a free interpretation of this source is as follows.

The whole available corpus must be divided into two parts: a training set and a test set. The training set is usually larger and it is used for estimating the parameters. In the case of the speech processing the parameters would be the probabilities in the *n*-gram language model. The quality of the *n*-gram language model should be measured on some unknown text or speech. We can use for this purpose the test set that is independent of the previously computed LM. A usual measure of the closeness of the *n*-gram language model to the speech or text in the same language is the perplexity that is introduced in Chapter 6.4.11. Sometimes the test set is used for the selection of the best language model. The test set can also contain the transcriptions of the speech data on which some other parameters of the recognizer are tuned. In this way the test set becomes not so independent. In this case such test set should be called a development test set (another term used in the speech recognition field is a devtest set). The final results should be evaluated on some other data – the true test set.

The test speech databases used for the evaluation of our speech recognition experiments are such independent test sets. In the course of the development of our recognizer also development test sets were used. Whenever the results using such dependent data are reported in this thesis, this will be indicated.

6.1.5. A Brief Methodology and Terminology Regarding the Counting of Words in the Text Corpus

The result of counting the words in the corpus is the list of the words and their frequencies. The methodology of counting the words in the corpus may vary according to the answers to at least the following two questions:

1. Are the words converted to the uniform case before their counting?
2. Are punctuation marks counted as words?

Regarding Question 1, the practice of SpeechLab is to convert each letter to the lower case before counting the bigram LM, because the case of the letters has no influence on the pronunciation. But we want the output of our recognizer to be as close to the correct transcript as possible. That is why it should include punctuation and correct placement of upper case letters. For this reason we have also counted the frequencies of words with their original case. In this way we can find out the most probable case of each word and the words in the vocabulary of our recognizer are written using this case. The case of many words depends on context. For example the word *český* in the context *český jazyk* (*Czech language*) is an adjective that should begin with lower case, and in the context *Český Dub* it is a part of a place name [51]. It means that a more precise method of transcription with correct letter cases would be to use some *n*-gram language model (possibly a bigram LM computed from words with their original case) which would be applied during a post-processing stage of the recognition. [13]

Regarding Question 2, punctuation marks are not counted as words in the corpus statistics published in this thesis. But our corpus is used also for part-of-speech tagging. The tagging could help to identify the beginnings of sentences where a capital letter should always be placed. We should also be able to place the proper punctuation (periods and commas) inside the sentences during a post-processing stage of the recognition. For this purpose another special bigram LM should be used, and so the punctuation in our corpus is surrounded by an extra space character to enable its computation.

Regarding the names for the units that are counted in the corpus, various authors may use a different terminology. The one that is used in this thesis is drawn from [6] (pp. 193 – 196). The units that we count in our corpus are called wordforms. The **wordform** is the inflected form of a certain lemma. The **lemma** is a set of lexical wordforms having the same stem and the same word sense. The size of the vocabulary of the corpus is the number of its wordform types. The total number of running words in the corpus is the number of its **tokens**. Using this terminology, we can say that our recent corpus has 2,099,353 wordform types and 360,104,333 tokens. (See Chapter 6.1.3.) The purpose of the term "wordform" is to emphasize the inflective nature of the language. The term "word" whenever used in this thesis has usually the same meaning as the term "wordform".

While this chapter deals with what to count in the text corpus, Chapter 7.1.2 suggests how to efficiently count words and successions of words in it.

6.2. Vocabulary

6.2.1. The Purpose of the Vocabulary for the Recognizer

The recognizer classifies the input stream of the speech signal into a succession of words that are present in its vocabulary (equivalent terms are "lexicon" or "dictionary"). Each word in the vocabulary is linked to its acoustic model represented by HMM via its phonetic transcription. Besides HMMs the recognizer also uses the information about the probability that a certain word will be uttered on condition that a string of certain words has been uttered before. This information is present in the language model. A type of LM studied in this thesis is the so-called bigram LM. It is a table of conditional probabilities that a certain word will be uttered on condition that a certain word had been uttered before. The words in the LM are only those that are present in the lexicon. If the lexicon changes, the LM must be recomputed. See Chapter 3.2 for information about how the lexicon, HMMs, and LM work together in the recognizer studied in this thesis.

6.2.2. A Brief Theory and Terminology of Lexicons for Automatic Speech Recognition

The most popular method of selecting words to populate the recognizer's vocabulary is to compute a list of words and their frequencies in some large text corpus and then select the most frequent words. The resulting vocabulary does not guarantee to contain every word in the speech to be recognized. Such a lexicon is called an **open vocabulary** and it is the only possible kind of vocabulary for the recognition of speech being used at present, e.g. the latest broadcast news.

In the initial stages of the development of some recognizer the vocabulary must be usually very small, e.g. containing few hundreds of items. To ensure that such a small vocabulary contains a reasonably large number of words in the recognized speech, the words from this speech, which usually belongs to the so-called development data (see Chapter 6.1.4), are put into the vocabulary. Sometimes only those words that constitute the development set are present in the lexicon, and sometimes also some other words are added to make the recognition task more demanding. In both cases such a lexicon is called a **closed vocabulary**.

A lexicon containing several thousand items, e.g. a 20,000-word lexicon, is sometimes briefly called a "20k lexicon", the "k" standing for kilo (thousand).

The quality of vocabularies can be measured without performing ASR by means of an indicator called the "out-of-vocabulary rate" often abbreviated as the OOV rate. The OOV rate is computed according to formula (3) and its result is in percent.

$$OOV\,rate = \frac{OOV}{S} \cdot 100 \qquad (3)$$

The *OOV* value in equation (3) is the number of tokens in the test corpus that have not been found in the vocabulary. The *S* value is the number of all tokens in the test corpus. Sometimes the OOV rate is expressed in terms of the so-called "coverage" computed according to formula (4).

$$Coverage = 100 - OOV\,rate \qquad (4)$$

6.2.3. Selecting Words for the Vocabulary Using the Analysis of the Corpus

The process of a compilation of the first open vocabulary for our continuous speech recognizer is described in article [54]. The size of this vocabulary was 20,000 items. Such a size is much too small for practical applications of the recognition of in advance unknown speech. We just wanted to make our first theoretical study of the implementation of the open vocabulary into our recognizer, from which we could move ahead towards larger lexicons. To improve the coverage of this relatively small vocabulary we decided not to apply only the frequency approach (selecting the first 20 thousand most frequent words in the corpus with the exclusion of some unsuitable words). We have also applied some more intelligent solutions like analyzing the grammatical cases of the most frequent words and supplying some important words in frequent grammatical cases into our vocabulary even if they did not appear among the 20 thousand most frequent words in the corpus. Another intelligent solution was a comparison of two corpora of quite different contents (topics and writing style) to get a set of words that are present in both corpora, which means that such words are not topic-dependent. Comparison of our lexicon with this set should lead to an improvement. One of the pitfalls of the frequency approach is the possibility that the source corpus is biased to some prevailing topic. This is relevant in particular to relatively small corpora. In the time when we were doing this research our corpus was six times smaller than our current corpus introduced in Chapter 6.1.3. The details about the compilation of our 20k lexicon are shown in Chapter 7.2.1.

Our next project following the 20k lexicon was the compilation of an 800k lexicon for our dictation system. The dictation system requires its user to dictate single words. After uttering each word the dictation system writes down its text form. This is called discrete speech recognition, while the subject of this thesis is continuous speech recognition. Since the discrete speech recognition is a far simpler task than CSR, see Chapter 2, the language model that supports it can also be simpler. A simpler LM than a bigram LM is the so-called unigram LM. It is just a table of probabilities of each word in the lexicon based on the frequency of this word in the training corpus. It has as many values as the number of items in the lexicon, while the number of values in the bigram LM is equal to the number of items in the lexicon squared. The unigram LM containing 800k probabilities for the 800k words in the lexicon can be implemented in today's PCs. The implementation of a bigram LM for the 800k lexicon is so far unfeasible. The answer to the begging question what this 800k lexicon has to do with CSR is the following: We wanted to make a robust (i.e. assessed on a large corpus) estimation of coverage of lexicons of various sizes for the case of the Czech language. The coverage is the most important parameter of the speech recognizer provided all its components work well. For example, the improvement of LM would not help the recognition when some frequently used words were missing in the vocabulary.

In contrast to the compilation of the 20k vocabulary, the creation of the 800k lexicon using the corpus was based purely on the frequency approach augmented by utilizing the spellchecker [55] built in the Czech version of the MS Word editor and by manual editing, i.e. reading the most frequent words and judging their usefulness according to personal knowledge. The most frequent words rejected by the spellchecker should also be manually checked for useful words for the vocabulary. The set of words rejected by the spellchecker actually contains modern words used in the newspapers and the names of prominent people. All in all the analysis of a corpus with the purpose to obtain a good lexicon (i.e. a lexicon with the smallest number of words and at the same time with the largest possible coverage) is a very laborious process. Chapter 6.2.8 shows more exact schemes of lexicon creation and updating according to the collected corpus.

6.2.4. Selecting Words for the Vocabulary Using Word Synthesis

To avoid the laborious process of vocabulary creation using the statistical analysis of the corpus and the subsequent manual editing we have tried to get a set of words for our vocabulary using the so-called word synthesis. This approach was based on generating all possible wordforms using a set of Czech stems, endings, and generation rules, all provided by the Institute of Formal and Applied Linguistics of the Charles University in Prague [52]. Using this approach we were able to synthesize up to 1.2 million different wordforms. Unfortunately, many of them were just hypothetical forms that are rarely seen in meaningful texts. This is why we put some stronger limits on the generating rules by excluding some archaic and informal words, and disallowing negative forms of some verbs and adjectives. This resulted in a set of some 600,000 different wordforms. Unfortunately, quite a lot of words occurring in modern Czech, and particularly many frequent proper names were missing in this inventory. [56]

The most serious disadvantage of the automatic generation of words is the fact that the result contains no information about the probability that the word could be used in text or utterance. This information is needed for the creation of the language model. [12]

The result of the statistical analysis of the training corpus described in Chapter 6.2.3 was finally merged with the result of the word synthesis described in this chapter to obtain the final vocabulary of 800 thousand words. The results of the coverage of independent corpus by this lexicon are given in Chapter 7.2.2.

We have also made a comparison of the coverage of the lexicon compiled using purely the statistical analysis of a corpus and the lexicon containing only the synthesized words. The result is that the statistical analysis of a corpus and subsequent manual annotation of the most frequent words can produce smaller vocabularies with larger coverage than the collection of words generated from a list of few prefixes, stems, and suffixes, (these three components are called morphemes), and the rules of morphotactics, i.e. the model of morpheme ordering that explains which classes of morphemes can follow other classes of morphemes inside a word [6] (pp. 65 and 86). [12] See Chapter 7.2.2 for details, especially Table 13.

6.2.5. Productivity of the Czech Language

While recognition systems designed for English can work well with a lexicon containing some 30 – 40 thousand words, the Czech language needs lexicons at least 10 – 20 times larger. This is caused by the very complex Czech morphology, namely its inflectional nature, which allows the nouns, pronouns, adjectives, numbers and verbs to take many different forms according to the grammatical context. [56]

Czech morphology includes besides 7 cases (nominative, genitive, dative, accusative, vocative, locative, instrumental), 3 genders (masculine, feminine, neuter), 2 numbers (plural, singular) of nouns, pronouns and adjectives, 3 tenses (past, present, future) and 2 voices (active, passive) of verbs also negation of many words, their grade, and diminutives (e.g. *star*: *hvězda – hvězdička*, *piece*: *kus – kousek*). There are also diminutive wordforms that have no equivalent in English, like *malý – maličký – malilinký* (*small*). In contrast to English, negation and grading of words in Czech produce new words (e.g. *difficult – more difficult – the most difficult*: *těžký – těžší – nejtěžší*). Czech surnames and terms for most professions differentiate between genders (e.g. *Novák – Nováková*, *teacher*: *učitel – učitelka*). Like in English, Czech words can have besides standard prefixes for negation and grade many other prefixes, for example *anti*, *euro*, *mega*, *sub*, *super*, *vice*. Czech verbs can have 20 possible prefixes that modify their meaning. Spoken spontaneous Czech that can also be found in broadcast news is richer in verbs with unusual prefixes than written language. This means that Czech language is very productive in the sense that many words can be derived from a single root. According to regular rules rare but grammatically correct words can be easily formed. [12]

6.2.6. Homonymy and Synonymy

In the course of manual annotation of our lexicons we had to deal with problems that are related mostly to the linguistic phenomenon of homonymy and synonymy. The very identification of these problems had arisen from many hours of experimenting with the prototypes of our recognizers. [12] Most of these problems could be solved only by very laborious manual annotation of our vocabulary.

Homonymy is defined as a relation that holds between words that have the same form with unrelated meanings. The items taking part in such a relation are called **homonyms**. [6] (p. 592) Homonyms that have both the same spelling and pronunciation cause no problem in ASR, because the task of ASR usually does not comprise word sense disambiguation. But special categories of words that can be counted among homonyms called homophones and homographs must be treated in a special way during the vocabulary creation.

Homophones are words with the same pronunciation but different spellings. [6] (p. 593) The examples of English homophones are the words *would – wood*, or *be – bee*. The examples of Czech homophones are *jet* (*to ride*) – *jed* (*poison*). In this example the phonetic phenomenon of the so-called assimilation causes the words *jet* and *jed* to sound the same in some contexts. The assimilation is a change of sound of a consonant according to the neighboring consonant, so that the whole group of consonants was either voiced or unvoiced [58] (p. 145). Homophones can be of two kinds: homophones of different meaning and homophones of the same meaning. The abovementioned homophones have a quite different meaning. Such homophones when they are in the recognizer's vocabulary have the same phonetic transcription. The right choice of the homophone by the recognizer depends on the language model that must take into account the context of words. A special kind of homophone are words that differ in the case of letters. Many Czech place names (e.g. *Ústí*) and surnames (e.g. *Donutil*) have their homophone counterpart written in lower-case letters. Our current vocabulary has these homophones written in their most frequent orthographic form. There is also a possibility of a special bigram LM that determines the right case of words in the phase of post-processing. This problem is also discussed in Chapter 6.1.5. Article [51] mentions another class of very frequent Czech homophones. These are verbs in plural number and past tense. Their spelling differs in dependency on the gender they refer to. For example the plural form of the Czech equivalent of the English verb *to be* in the past tense is *byli* for the masculine gender and *byly* for the feminine gender. This frequent verb has also a homophone counterpart *bili*/*bily* which is a plural past tense form of the verb that means *to beat*. From the phonetic point of view homophones are also some pairs of words and sequences of words, e.g. *NATO* (*North Atlantic Treaty Organization*) – *na to* (*on this*). The right choice of these words again depends on the LM. The other kind of homophones are words with the same meaning and different orthographic (written) variants. Such words are called synonyms and they are discussed later in this chapter.

There are also pairs of words with identical orthographic forms and different pronunciations. These words are called **homographs**. [6] (p. 593) The example of English homograph is the word *bass* that can have meanings associated either with fish or music. The example of Czech homograph is the word *byty* that means either *flats* or *bytes*. Homographs are represented in our vocabulary by a single lexical unit that has more than one phonetic transcription.

Synonymy is defined as a relation that holds between different words that have the same meaning. Two words have the same meaning if they can be substituted for one another in a sentence without changing either the meaning or the acceptability of the sentence. [6] (p. 598) The examples of English synonyms are the words *big – large*. Czech synonyms that have a different both orthographic and phonetic form, e.g. *slovník – lexikon*, are not subjects of special editing in our vocabulary with the exception of the synonyms that have only a slightly different orthographic and phonetic form, e.g. *pasivní – pasívní* (*passive*). Such synonyms and the synonyms that are at the same time homophones, e.g. *ateismus – ateizmus – atheismus*, or *zčistajasna – z čistajasna – zčista jasna – z čista jasna* (*out of the blue*), must be rewritten to their most frequent variant of spelling, so that they have only a single representative in our vocabulary. This rewriting process is a part of corpus cleaning, see point number 8 in Chapter 6.1.2. From the phonetic point of view synonyms are also some groups of homographs that have a different pronunciation. In Czech language the most frequent reason for a different pronunciation of the same word is the abovementioned assimilation. Because of assimilation many words in our vocabulary have multiple phonetic transcriptions. Since the rules of the assimilation in the Czech language are relatively simple, the additional phonetic transcriptions could be added automatically. Another resource of the information about the additional phonetic transcriptions is the list of wordforms that should be rewritten to their standard orthographic variants. This applies e.g. to the abovementioned pair of words *pasivní – pasívní*, where a different length of a vowel *i – í* indicates a possible duality in pronunciation.

6.2.7. Lexicon Structure

The lexical item of vocabulary for all kinds of speech recognition tasks solved in our lab (continuous speech recognition, dictation of discrete words) is the wordform, i.e. a word that has a certain grammatical category. The alternative to such representation could be the situation in which the vocabulary would contain word roots and possible prefixes and suffixes. The advantage of such representation would be the smaller size of the resulting lexicon. The disadvantage would be the need to capture longer sequences of such lexical items in the language model, which means that such a model would need a relatively large amount of computer memory.

The recognizer aligns words to acoustic speech signal, which means that each word in its lexicon must have its phonetic transcription. The phonetic transcription is a concatenation of phonetic units which are text characters representing acoustic units of speech, the so-called phonemes. Chapter 6.3 gives details about automatic phonetic transcription.

There could be more information than only a phonetic transcription linked to the word in the recognizer's vocabulary. In article [13] we propose a structure described in Table 1. [51] The lexicon with this structure can also be used for corpus cleaning (point 8 in Chapter 6.1.2) and for corpus tagging. Phonetic transcriptions in the column "Pronunciation" are ordered by the subjectively evaluated probabilities of their use in human utterances. In this way we can easily remove some less probable pronunciation variants from the vocabulary when we want to speed up the recognition or test the importance of the additional phonetic transcriptions.

There can be maximally 8 phonetic transcriptions of a single word in our current version of the 312k vocabulary and recognizer. [15]

Table 1. Lexicon structure shown on examples of 3 words [13]

Standard Orthography	Alternative Orthography	Pronunciation	Morphology Class
million	million	milijon, milijón	Num1
téze	these, teze	téze, teze	Noun1
s		s, z, sE, zE	Prep4, Prep7

6.2.8. Lexicon Updating

As already mentioned at the end of Chapter 3.1, our largest vocabulary for CSR contains 312,000 wordforms. We have obtained this vocabulary mainly by statistical analysis of our gradually increasing corpus. This vocabulary had its 20k, 140k, and 200k predecessors, each built on the ground of the smaller one. We have not used our 800k vocabulary introduced in Chapter 6.2.3 for the derivation of the 312k vocabulary, because it was tailored too much to our discrete speech dictation system.

In the course of several processes of lexicon and training corpus enlargements we have developed procedures for these two cases. The process of compilation of initial vocabulary is depicted in Figure 2. The process of lexicon updating from a new part of the training corpus is shown in Figure 3 and Figure 4. Figure 3 shows the process of removal of the obsolete words from the already existing lexicon provided a new part of corpus has been collected. These obsolete words are not correct according to the spellchecker and not present in the new corpus but they used to be regarded as useful in the past. The updated lexicon in Figure 4 is put together from the former lexicon without obsolete words, the list of words correct according to the spellchecker, and the list of words incorrect according to the spellchecker but approved by humans.

The threshold in Figure 2 and Figure 4 should be stated in dependence on the target size of the compiled lexicon and so that the number of words for the checking by humans is feasible. It is useful to keep the list of the words that are approved by the spellchecker but that are not regarded as useful for the recognition task. In Figure 2 this list is compiled for the first time, and in Figure 4 it is put together from the former list and the list of new such words found in the new training corpus. The other lists *a* to *o* in the figures do not need to be remembered. Their content can be derived from their path in the oriented graph. The list of words in the corpus changes whenever the new part of the training corpus is added to the original training corpus after the updating of the lexicon.

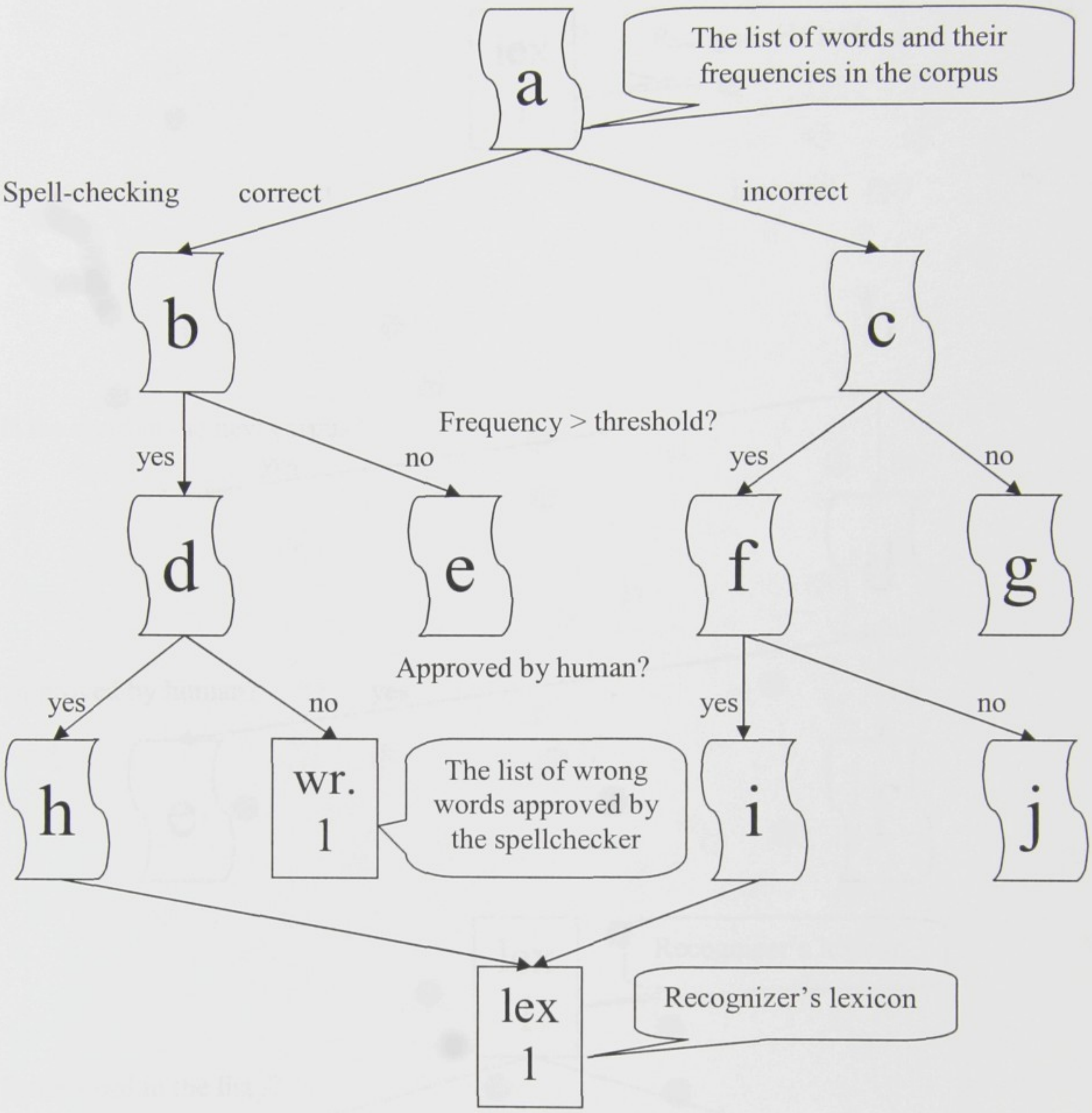


Figure 2. Compilation of the first lexicon for the recognizer

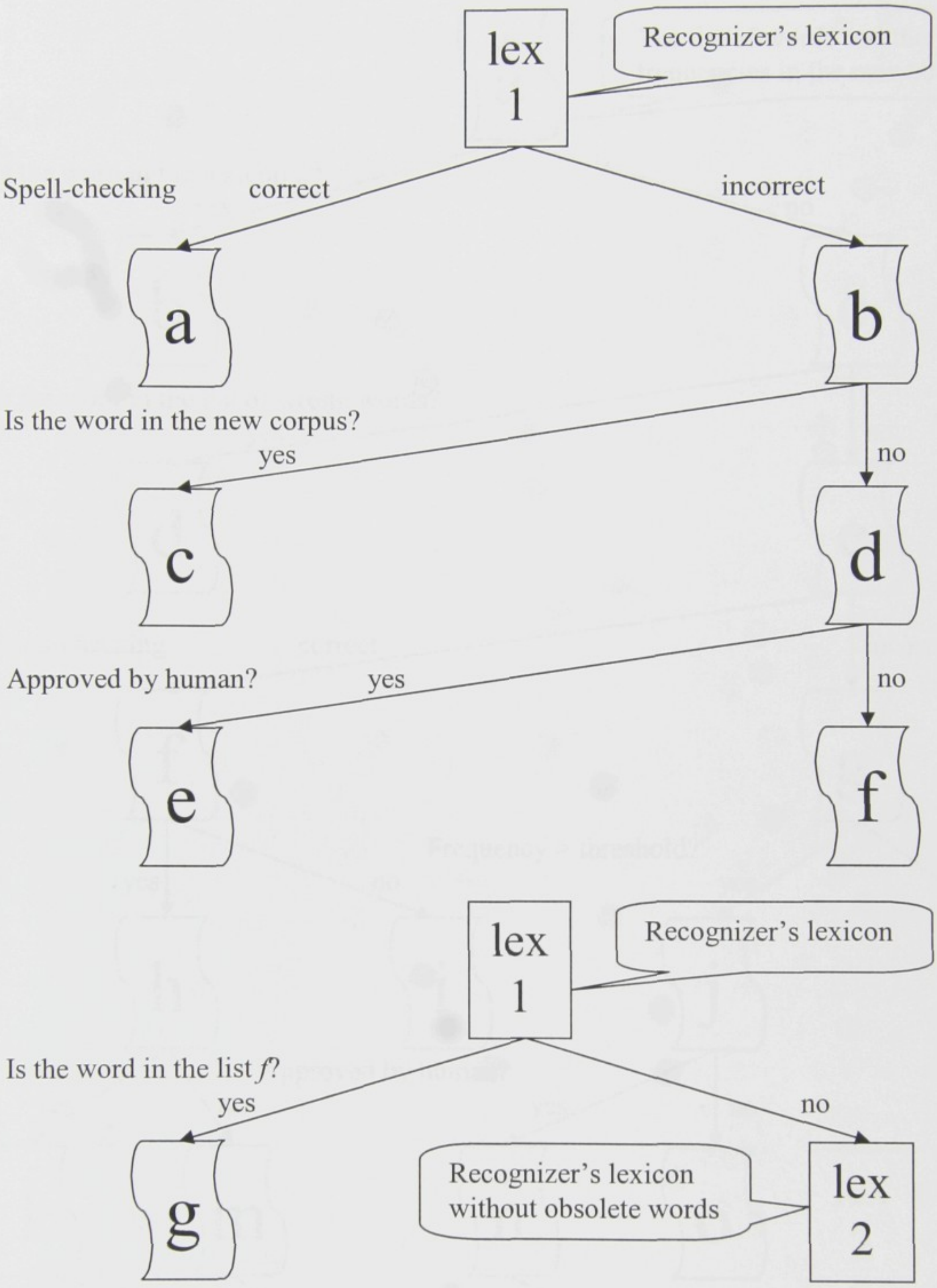


Figure 3. Removal of the obsolete words from the lexicon using the new part of the training corpus

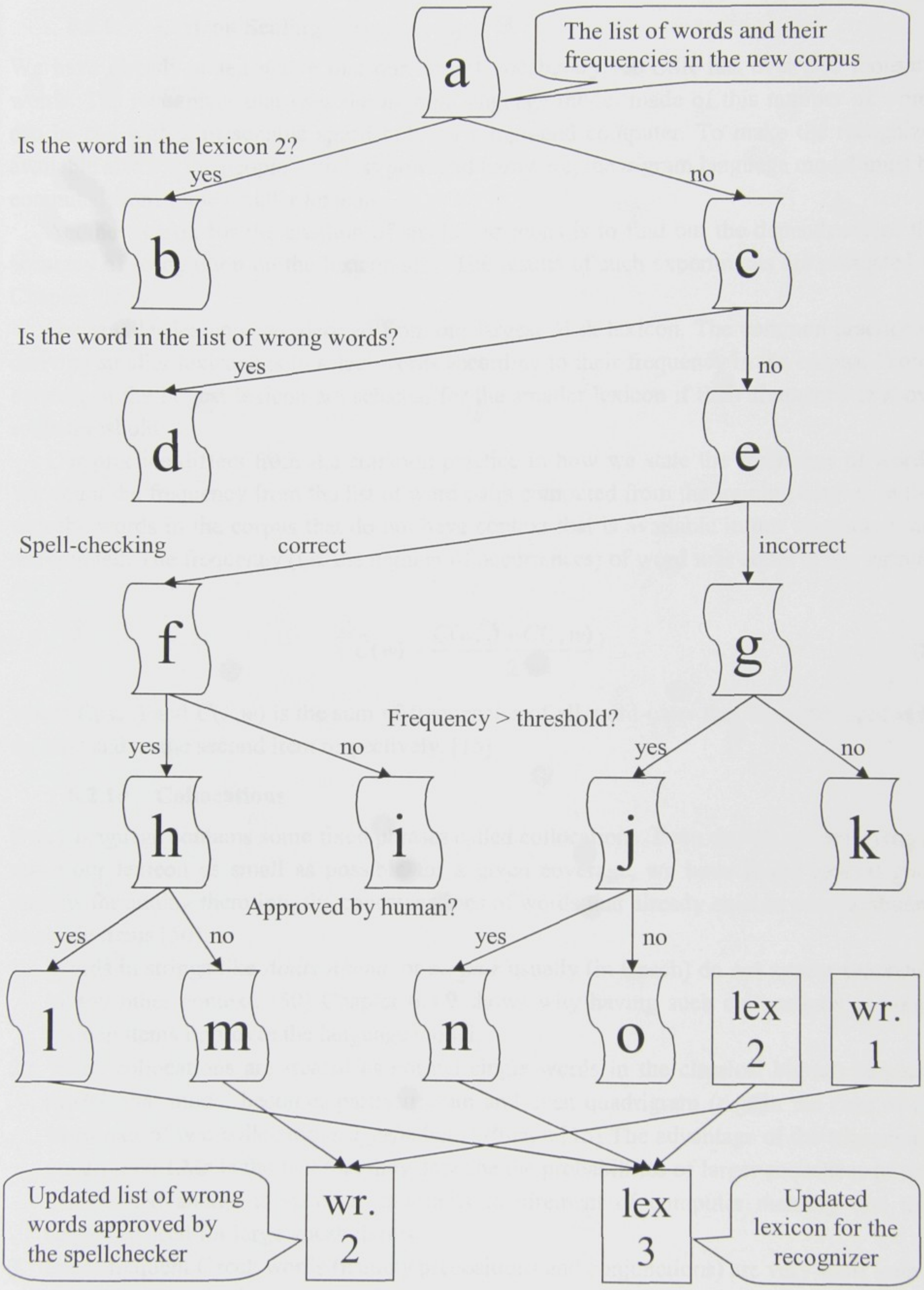

a
The list of words and their frequencies in the new corpus
Is the word in the lexicon 2?
yes
no
b
c
Is the word in the list of wrong words?
yes
no
d
e
Spell-checking
correct
incorrect
f
g
Frequency > threshold?
yes
no
yes
no
h
i
j
k
Approved by human?
yes
no
yes
no
l
m
n
o
lex 2
wr. 1
Updated list of wrong words approved by the spellchecker
wr. 2
lex 3
Updated lexicon for the recognizer

Figure 4. Updating of the lexicon using the new part of the training corpus

6.2.9. Lexicon Scaling

We have already stated above that our largest vocabulary for CSR has over 300 thousand words. The recognizer that uses the *n*-gram language model made of this number of words can be run with a reasonable speed only on a high-end computer. To make the recognizer available also for the people with less powerful hardware, the *n*-gram language model must be computed from some smaller lexicon.

Another reason for the creation of smaller lexicons is to find out the dependency of the accuracy of recognition on the lexicon size. The results of such experiments are presented in Chapter 7.2.4.

Our smaller lexicons are derived from our largest 312k lexicon. The common practice of deriving smaller lexicons is to select words according to their frequency in the corpus. Words existing in the largest lexicon are selected for the smaller lexicon if their frequency is above some threshold.

Our practice differs from the common practice in how we state the frequency of words. We count the frequency from the list of word-pairs computed from the training corpus. In this way the words in the corpus that do not have context that is available in our vocabulary are not counted. The frequency (i.e. the number of occurrences) of word *w* is equal to the formula (5)

$$\widehat{C}(w) = \frac{C(w,.) + C(.,w)}{2} \qquad (5)$$

where *C*(*w*, .) and *C*(. ,*w*) is the sum of frequencies of all word-pairs that have the word *w* as the first and as the second item respectively. [15]

6.2.10. Collocations

Every language contains some fixed phrases called collocations. Even though we are trying to make our lexicon as small as possible for a given coverage, we have found several good reasons for adding them into the concatenations of words that already exist in our vocabulary as single items [50]:

1. Words in strings like *Addis Abeba*, or *au pair* usually (in Czech) do not appear separately in any other context. [50] Chapter 6.4.9 shows why having such collocations as single lexicon items improves the language model.
2. When collocations are treated as normal single words in the classical bigram language model, that model becomes partly trigram and even quadrigram (e.g. in the case of the word-pair of two collocations *a_zejména v_Praze*). [15] The advantage of the trigram and quadrigram LMs is the fact that they describe the probabilities of larger sequences of text. Their disadvantage is such an enormous requirement of computer memory that they cannot be used for large vocabularies.
3. Some frequent Czech words (namely prepositions and conjunctions) are very short (one or two phonemes) and when pronounced together with the following word, they are often omitted by the recognizer. [50] The examples of collocations that solve this problem are *v_Praze* (*in Prague*), *v_Brně* (*in Brno*). [15]
4. Collocations with prepositions and conjunctions can also be better phonetically described. For example, thanks to the phenomenon of the so-called assimilation, mentioned in

Chapter 6.2.6, the collocation *v_Praze* should be pronounced [*fpraze*] whilst the collocation *v_Brně* should be pronounced [*vbrňe*]. [15] Some concatenations of words cause a big difference in pronunciation of these words. For example *devatenáct_set* (*nineteen* as a part of a year in the date) should be pronounced as [*devatenácet*] and not as [*devatenáct set*]. [50]

When the lexicon contains a word that is a part of a collocation whose other components are missing in the lexicon, for example the lexicon contains the word *Burkina*, but does not contain the word *Faso*, we can identify the incomplete collocations by simple finding such words that have zero wordform types of predecessors or successors in the list of word-pairs computed from the training corpus and the lexicon. This allows us to improve both the lexicon and the language model. We have identified collocations using the standard statistical tests recommended for this purpose in [59] applied on the list of word-pairs: [51]

1. **Chi-square (χ^2) test** gives the highest score to the collocations composed of relatively frequent words that have very few types of complement words, e.g. *Buenos_Aires*, *Rolls_Royce*.
2. ***t* test (Student's test)** gives the highest score to the collocations composed of very frequent words that have diverse neighborhoods, e.g. *já_jsem* (*I am*), *více_než* (*more than*).
3. **Pointwise Mutual Information (PMI)** retrieves the collocations composed of rare words with few types of complement words, e.g. *kapalným_vodíkem* (*liquid hydrogen*), *fackovacího_panáka* (*whipping boy*).

Each statistical test gives each word-pair a certain value. The word-pair is considered to be a collocation if this value is higher than some threshold. The values are computed according to the formulas (6) to (8). These formulas are composed of the following operands:

WordPairCount = the number of occurrences of the word-pair in the training corpus.

Word1AverCount = the average number of occurrences of the first word in the word-pair in the training corpus computed according to the equation (5).

Word2AverCount = the average number of occurrences of the second word in the word-pair in the training corpus computed according to the equation (5).

FollowingTokens1 = the number of tokens following the first word in the word-pair.

PrecedingTokens2 = the number of tokens preceding the second word in the word-pair.

n = the total number of tokens in the training corpus.

b = *PrecedingTokens2* – *WordPairCount*

c = *FollowingTokens1* – *WordPairCount*

d = *n* + *WordPairCount* – *FollowingTokens1* – *PrecedingTokens2*

h = *WordPairCount* * *d* – *b* * *c*

$$\chi^2 \textbf{ test} \text{ value} = n \,/\, PrecedingTokens2 \,/\, FollowingTokens1 \,/ \\ /\, (n - PrecedingTokens2) \,/\, (n - FollowingTokens1) * h * h \tag{6}$$

$$\textbf{\textit{t} test} \text{ value} = (WordPairCount - Word1AverCount \,/\, n * Word2AverCount) \,/ \\ /\, \text{sqrt}\,(WordPairCount) \tag{7}$$

$$\textbf{PMI} \text{ value} = \log_2 (n \,/\, \mathit{Word1AverCount} \,/\, \mathit{Word2AverCount} * \mathit{WordPairCount}) \qquad (8)$$

The operands in the formulas (6) to (8) are ordered in such a way that does not cause any arithmetic and floating-point operation overflows in the course of the computation of intermediate values on the computer.

We have both concatenated collocations and their original components in the language model and vocabulary, which seems to be redundant for the type of collocations like *Buenos_Aires*, but we need their component words in the vocabulary for the correct evaluation of word error rate. Concatenation of words into collocations is the most valuable in the case of the shortest words, which are often uttered unclearly. [51]

There are about 1,700 collocations in our latest 312k vocabulary. They can be recognized as words concatenated by the character "_".

6.2.11. Compound Words

Compound words are concatenations of several words. Czech is not as rich in compound words as for example German language, but the number of compound words in Czech language is significant enough to deserve a special treatment in language modeling. [51] There is a large class of compound quantitative adjectives of the type *patnáctiletý* (*15-year old*), *devítitunový* (*9-ton*), etc. that are causing nearly a combinatorial explosion in the Czech vocabulary. These words can be represented by a set of their components (*patnácti*, *devíti*, *letý*, *tunový*, …) and a set of rules for their concatenation into correct words. We have found that 686 parts of such words can generate 58,892 possible wordforms expressing various quantities in various grammatical cases. If we incorporate some splitting rules into our lexicon, i.e. replace the abovementioned compounds with their components and keep the list of all possible compounds apart from the lexicon, we could get a smaller lexicon that would have a larger coverage of the texts. [15]

The most frequent prefixes in the Czech language are *ne-*, *nej-*, and *nejne-*. The first prefix makes negations of verbs, adjectives and adverbs, and the other two prefixes make superlative forms of adjectives and adverbs. We have found in our corpus 60,000 wordforms that have these prefixes but that can also exist without them. This search involved the use of a Czech spellchecker [55] and a lot of manual editing because the spellchecker approves many words that are not really used in Czech language. For example when testing the word *nerostné* (*mineral*) the spellchecker approved the word *rostné* but this word is in fact a nonsense.

The resulting lexicon that contains components of words instead of the compound words described above is smaller by 21,000 words than the original 312k lexicon. Its OOV rate has dropped only by 0.05% in comparison to the original 312k lexicon. The decomposition of words is important mainly in smaller vocabularies. For example, the OOV rate of a smaller 64k lexicon has dropped by 0.33% when this lexicon was decomposed. [15]

Chapter 6.4.10 contains details about language modeling when some words in the lexicon are replaced with their components.

6.3. Phonetic Transcription

6.3.1. The Purpose of Phonetic Transcription

Phonetic transcription expresses the sequences of the sounds of spoken language by the sequences of the text characters. There are two main fields of the communication of humans with computers where phonetic transcription is applied. The first field is the synthesis of speech according to the text input. One possible application of speech synthesis is software for the computers for the blind people and information services via telephone line. The second field is automatic speech recognition. In both cases the characters of the text form of the language (graphemes) must be mapped to the characters that stand for the sounds that are usually uttered when this text is pronounced (phonemes). Phonetic transcription serves as a link between the text form and the acoustic models of words in the both fields.

Phonetic transcription is used twice in both speech synthesis and recognition. First, a training speech database must be rewritten to its text form and this text form must be phonetically transcribed. The result is used in the process of segmentation. In the course of segmentation the speech signal is divided into segments that contain a single phoneme. These segments are then used for the training of the acoustic models of phonemes. The initial stages of the segmentation must be manual. When more reliable acoustic models are trained, the segmentation can become more or less automated. Second, the phonetic transcription is applied to the text that should by uttered by the computer, and in the case of ASR the phonetic transcription of the recognizer's dictionary must be done.

6.3.2. The Units of the Sounds of Speech

The scientific field of phonetics can for a given language determine a set of sounds whose sequences can form every word of this language. The resulting description of speech can have various levels of fidelity. The most accurate phonetic transcription can be attained with a large set of sound units. The large number of units complicates both the phonetic transcription and the process of speech synthesis or ASR. The overview of various sound units used in various laboratories for the description of speech is published for the Czech readers in [60] (pp. 94 – 96).

SpeechLab uses the so-called phonemes for the sound description of speech. A **phoneme** is a unit that can discern the lowest number of speech sounds. The researchers in the field of phonetics try to design the phonemes in such a way that they mostly correspond to the letters of the orthographic form of the words. This is possible in the case of the so-called phonographic languages. The Czech language belonging to the group of phonographic languages has the number of phonemes very close to the number of letters in the alphabet of its written form. The advantage of this way of speech description is the relative easiness of phonetic transcription and a low computational workload in the course of ASR. The disadvantage is a lower fidelity of the description of speech.

The Czech language has a very transparent correspondence between the letters of the text form and the set of Czech phonemes. The infidelities of the phonetic transcription are caused by the fact that the sound shapes of the letters in the uttered words are influenced by the preceding and the following sounds. There are two ways how to solve this problem. The first way is to increase the set of the phonetic units. We can for example have phonetic units corresponding to the all possible syllables in the language. This approach is mentioned at the beginning of this chapter. The second way, which is adopted in the SpeechLab, is to train the acoustic models of the phonemes on the very large database of speech that is segmented into the phonemes in various contexts. The resulting acoustic models corresponding to the phonemes become more robust. More details about SpeechLab's practice regarding this topic can be seen in Chapter 3.2.

6.3.3. The Phonetic Alphabet

The phonetic alphabet is a set of characters used for the transcription of the sounds of words of a given language.

In 1886 the first variant of the International Phonetic Alphabet (IPA) was created. Its latest version was published in 1989 in [61]. It provides a set of characters based on the letters (graphemes) of various national alphabets and a set of principles for transcription. IPA makes comparison of languages and explanation of pronunciation of various languages for various nations possible. It is suited best for the group of the Western European languages, not so much for the Slavic languages, which comprise also the Czech language. This is the reason why IPA is replaced with some national set of phonemes when international comparison is not the case. [58] (p. 37), [6] (p. 93).

Table 2. The Phonetic Alphabet for Czech (PAC) [63]

Number	Phoneme in plain Czech	Phoneme in PAC	Example	Number	Phoneme in plain Czech	Phoneme in PAC	Example
1	“a”	a	tát**a**	21	“m”	m	má**m**a
2	“á”	á	t**á**ta	22	“m”	M	tra**m**vaj
3	“b”	b	bá**b**a	23	“n”	n	ví**n**o
4	“c”	c	o**c**el	24	“n”	N	ba**n**ka
5	“dz”	C	le**c**kde	25	“ň”	ň	ko**n**ě
6	“č”	č	**č**ichá	26	“o”	o	k**o**lo
7	“dž”	Č	rá**dž**a	27	“ó”	ó	**ó**da
8	“d”	d	je**d**en	28	“p”	p	pu**p**en
9	“ď”	ď	**d**ělat	29	“r”	r	be**r**e
10	“e”	e	l**e**v	30	“ř”	ř	mo**ř**e
11	“é”	é	m**é**ně	31	“ř”	Ř	ke**ř**
12	“f”	f	**f**auna	32	“s”	s	**s**ud
13	“g”	g	**g**uma	33	“š”	š	du**š**e
14	“h”	h	a**h**a	34	“t”	t	du**t**ý
15	“ch”	X	**ch**udý	35	“ť”	ť	ku**t**il
16	“i” or “y”	i	b**i**l, b**y**l	36	“u”	u	d**u**še
17	“í” or “ý”	í	v**í**tr, l**ý**ko	37	“ú” or “ů”	ú	r**ů**že
18	“j”	j	do**j**at	38	“v”	v	lá**v**a
19	“k”	k	**k**upec	39	“z”	z	ko**z**a
20	“l”	l	dě**l**á	40	“ž”	ž	rů**ž**e

Article [62] proposes the phonetic alphabet for ASR of the Czech language. The original table with the set of the Czech phonemes in [62] contains also their closest symbols taken from two well-known phonetic alphabets for the English language and a version written using only the English alphabet. The members of the SpeechLab and some other Czech scientists use this set in the form that is shown in Table 2. Examples of phonetic transcription in this thesis are also written using this alphabet.

In the course of annotating our speech databases and recognition experiments various other speech and non-speech events, sounds, and noises have been classified and given special characters used for their phonetic transcription, see Table 3. Each time a new type of sound is declared being worthy of classification, its acoustic model should also be trained. It means that all segments of the speech training database that contain this sound must be found in the process of segmentation (see Chapter 6.3.1), which is partly manual.

Table 3. The phonetic alphabet of the SpeechLab for additional speech events, non-speech events, sounds, and noises present in the Czech speech databases

Phoneme character	Description
E	schwa (See [6] (p. 162). It appears at the ends of consonants uttered in the course of spelling in Czech. It is also used in the transcriptions of some foreign words.)
-	silence (In fact it is the background noise.)
0	glottal stop (In [62] it is mentioned as “glottal plosive” or “hit”. It is a short sound that can be sometimes heard at the beginnings of words that begin with vowels [15].)
1	click (a short sound like a lip smack or a mouse click)
2	long low noise
3	breath (mostly a breath-in)
4	longer loud noise (a car, music)
5	hesitation sound or filled pause (“uh” and “um”, see [6] (pp. 194 – 195).)
6	laughter

6.3.4. Phonetic Transcription Based on the Explicit Rules

The first rule-based system for automatic phonetic transcription was developed in SpeechLab in 1999. It is described in [64]. It used a set of rules proposed for the Czech language in [60] (pp. 101 – 106). These rules are in the form (9) shown in [60] (p. 99). Formula (9) means that if *C* precedes *A* and *D* follows *A*, then *A* should be rewritten to *B*. *A*, *C*, and *D* are strings of graphemes. *B* is a string of phonemes. If *C* or *D* is any string, it is represented as an empty string.

$$A \rightarrow B / C\_D \tag{9}$$

The list of rules must be sorted according to the length of the concatenation of the strings *C&A&D* and checked from the longest to the shortest strings in the course of the transcription. After the application of the longest rule that matches the beginning of the current part of the string of graphemes that should be rewritten the process starts over again beginning with the first grapheme after the sub-string *A*. The text beginning with this grapheme must match the concatenation of the strings *A&D*, and the preceding string of graphemes must match the string *C* in the longest matching rule.

The main problem of this method of phonetic transcription is that besides several tens of basic rules, hundreds of exceptions must be manually found. The program for phonetic transcription must at first check the list of exceptions, and when no applicable exception is found the list of rules is applied.

The exceptions should be found manually. They are usually in the form of longer strings than the set of rules. Their construction must ensure that no conflict among them exists. This is sometimes tricky as can be seen in the following example.

One of the basic rules is *n* → *ň* / _ *i*. In the Czech language many words contain a sub-string *ni* that should be pronounced [*ňi*], but there is a lot of exceptions to this rule mainly in the words of foreign origin. According to this rule the word *unikáte* (*you are running away*) is correctly transcribed to [*uňikáte*]. Many Czech words have the origin in the English word *unique*, like *unikát* (*a unique thing*), *unikátní* (*unique*) (these two lemmas have tens of inflections), and the sub-string *ni* in these words should be pronounced as [*ni*]. And here comes the problem: The word *unikát* is shorter than the word *unikáte*, so that an exception

n → *n* / u _ *ikát*

would incorrectly transcribe the word *unikáte* into [*unikáte*]. So, there must be some character that means the end of word inserted into the exception, e.g.

n → *n* / u _ *ikát*_.

Due to the existence of many inflections of the word *unikát*, like *unikátu*, *unikátem*, *unikáty*, *unikátů*, *unikátům*, *unikátech*, many slightly modified exceptions must be added:

1. *n* → *n* / u _ *ikátu*,
2. *n* → *n* / u _ *ikátem*,
3. *n* → *n* / u _ *ikáty*,
4. *n* → *n* / u _ *ikátů*,
5. *n* → *n* / u _ *ikátech*.

It should be noticed that there cannot be an exception

n → *n* / u _ *ikáte*

instead of the exceptions number 2 and 5, because such an exception would incorrectly transcribe the previously mentioned word *unikáte*. Moreover, the word *unikáte* is also one of the inflections of the lemma *unikát*. So, in certain contexts this word should be rewritten into [*unikáte*]. The good thing is that such a situation would be quite exceptional, because this word is in the vocative case (meaning that the subject identified by this word is being addressed). We solve such cases by the frequency approach (i.e. we prefer the most frequent pronunciations for the ambiguous words) without using the statistics of the word-pairs.

If we represent the exceptions in the same form (9) as the rules, there would be no reason to make a difference between the exceptions and the rules. All the rules could be in a file that would be used by a relatively simple program for the phonetic transcription. The alternation of the rules would be a question of the editing of the file with the rules without a need to compile the program again.

If the transcription system described above had contained all the rules existing in the Czech language, its accuracy would have been very close to 100%, because the ambiguous words are not very frequent in this task. The problem that should be solved is to find some methods of automated acquisition of a consistent set of all these rules. One of the helpful solutions is described in Chapter 6.3.6.

Provided we already have a set of rules, we can test their consistency in the following way: We should try to find and remove possible duplications in the concatenations of the strings *C&A&D*, and then we should use this set of rules for the transcription of a test set of words whose correct transcriptions are already known. During this transcription the records will be made about the cases when some rule has produced an incorrect output and how many times each rule was used. The rules that have produced any incorrect output and the rules that have not been used should be altered or removed.

The system for the phonetic transcription described in this chapter produces for each word in our vocabulary only a single phonetic transcription. In Chapter 6.2.6 we have mentioned that there are multiple phonetic transcriptions of some words in our vocabulary. We have found four reasons for a single word to have more than a single phonetic transcription:

1. The phonetic phenomenon of assimilation. (The pronunciation of words changes in dependence of the adjacent words.)
2. The word is a homograph. (Typically the word has some meaning in the Czech language and at the same time some other meaning in some foreign language but with a different pronunciation.)
3. The word has an alternative orthographic variant that is present in our normalization list mentioned in point 8 in Chapter 6.1.2. Some of these variants can also be a source of the alternative phonetic transcriptions as it is written at the end of Chapter 6.2.6.
4. The word has more than one way of pronunciation, because different people have different speaking habits. This applies both to the words of foreign origin (some people pronounce them in the foreign way and some other people in the Czech way) and to the Czech words. Book [58] (pp. 320 – 345) mentions many rules of the Czech orthoepy (standard pronunciation) with many alternative variants. But this source is not enough for our purpose because people pronounce some words in a way that is regarded as colloquial or slang. The best other source of the information about the alternative pronunciations is checking the errors in the output of the recognizer and listening to the speech data that were recognized with these errors. These errors usually have some acoustic reason.

It is possible to make the additional phonetic transcriptions caused by the reasons 1 and 3 automatically or semi-automatically, but the other types of the alternative phonetic transcriptions must be edited manually. The less important pronunciation variants of the less frequent and longer words should be omitted so that the search space of the recognizer would not be too large. Our largest 312k lexicon has 18,933 additional phonetic transcriptions, which is equal to 1,06 transcriptions per lexicon item [15]. The effects of the additional phonetic transcriptions and some other recognition experiments regarding phonetic transcriptions are given in Chapter 7.3.

6.3.5. The Use of Neural Network for Phonetic Transcription

A system for phonetic transcription based on an artificial neural network was developed in SpeechLab in 2000. It is described in [65] (in English), [7] and [63] (in Czech).

The system draws on the idea of the famous neural net called NETtalk published by the American scientists Terry Sejnowski from Johns Hopkins University and Charles Rosenberg from Princeton in [66] and [67]. This research is also briefly described in [3] (pp. 585 – 586). Our neural network reads five graphemes of the word to be transcribed in a sliding window and guesses the phoneme for the central grapheme. Two graphemes on either side of this central grapheme provide a context that helps to determine the pronunciation.

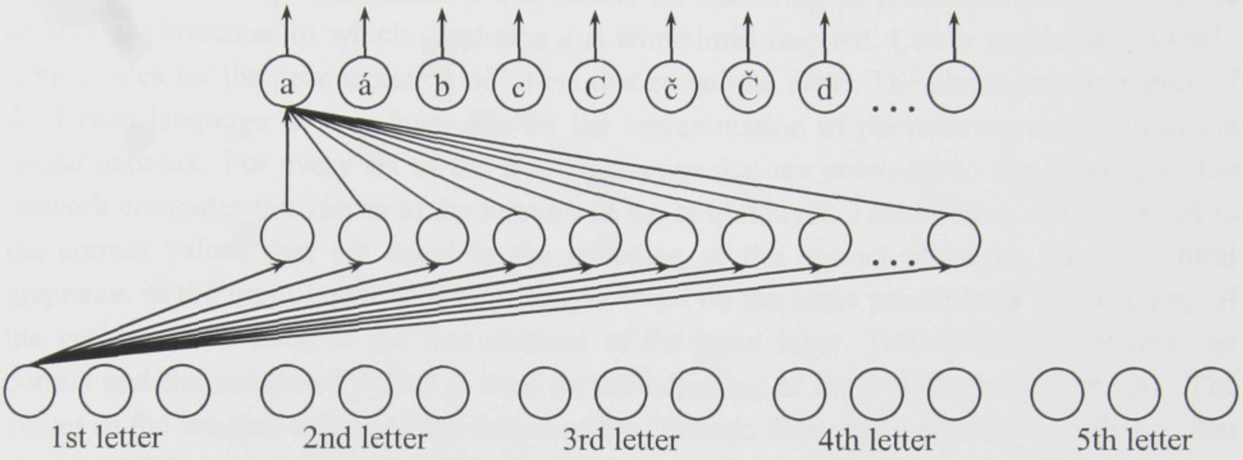


Figure 5. Neural net for phonetic transcription of the Czech language [63]

Figure 5 shows the scheme of our neural network. The neurons of the network are symbolized by the circles. The lower row of neurons is the input layer, the row above it is the hidden layer, and the upper row is the output layer of the network. The neurons in the adjacent layers are connected by the synapses in such a way that each neuron is connected with all other neurons in the adjacent layer. Each synapse is assigned with its weight. The weight is a real number.

The input layer has so many neurons as the number of graphemes in the Czech alphabet times five. We had 44 graphemes in the Czech texts including a special character that must fill the beginning and the end of the input words, so that their first and their last grapheme can appear in the center of the sliding window. So, the input layer has 220 neurons organized in five sections of 44 neurons. The five sections correspond to the five graphemes in the sliding window. A grapheme is encoded in its section as an excitation of one of the 44 neurons. Mathematically it means that the neuron that corresponds to the grapheme is assigned with number one and all the other neurons in the same section are assigned with zeros. So, in each step there are exactly five neurons assigned with number one and the other neurons are assigned with zero in the input layer.

The hidden layer has 56 neurons because the number of neurons in the output layer is also 56. In fact, the number of neurons in the hidden layer can be arbitrary. In one of our experiments we doubled the number of neurons in the hidden layer, and the performance of the network slightly improved, but the time consumed by the learning of the network increased considerably. It is also possible to have more than one hidden layer in the network.

The output layer has so many neurons as the number of phonemes. Our network had 56 output neurons. This number is higher than the number of phonemes in the PAC in Table 2.

The reason for this is the need to introduce some additional phonemes that allow mapping of single graphemes to single phonemes. (The rules in the rule-based system described in Chapter 6.3.4 could also map strings of graphemes to the strings of phonemes.) For example, the phonetic transcription of the word *Emma* is [*ema*]. One of the letters in the sub-string *mm* must be mapped to a special phoneme that means no sound. The phonetic transcription of the word *Lucie* is [*lucije*]. The sub-string *ie* must be mapped to the string of phonemes *ije*. So, we have introduced a special phoneme that stands for the string of phonemes *je* as one of the additional phonemes to which grapheme *e* is sometimes mapped. Czech words have mostly simple rules for the pronunciation of letters that constitute them. The phonographic nature of the Czech language is very favorable for the representation of phonetic transcription in the neural network. For every set of the five graphemes that are entered into the input layer the network computes the values of the neurons in the output layer. These values are compared to the correct values that are equal to the encoding of the correct phoneme for the central grapheme in the input layer. This encoding is based on the same principle as the encoding of the graphemes in each of the five sections of the input layer. The difference between the correct and the computed values is used for the adjusting of the weights in the network. The values of the weights influence the output of the network. They are gradually modified so that the error of the network is reduced.

The network was trained on the set of 5 thousand words and its performance was at the same time measured on the test set of some other 5 thousand words. These 10 thousand words were the most frequent words in our newspaper corpus that we had collected in 2000. The foreign words with irregular pronunciation, e.g. *Angeles* [*enČls*], were not present in these sets. The Czech words of foreign origin with exceptions in pronunciation, e.g. *unikátní* [*unikáťňí*] (*unique*), were a part of these sets. We had computed the phonetic transcription for each of these 10 thousand words with the use of our rule-based system developed in 1999 and manually corrected eventual mistakes. We had also converted the phonetic transcriptions to the form that enables mapping of single graphemes to single phonemes as it is described in the previous paragraph. The network has learned phonetic transcription on the set of 5 thousand words by the method of back-propagation that is described in many textbooks about artificial intelligence and neural networks, e.g. [3] (pp. 578 – 584). It is also presented in [63].

The result of this research was the following: Our neural network has learned the most of the Czech basic pronunciation rules and was able to apply these rules to the words that were not part of its training set. The problem was that the network was not able to learn all the basic pronunciation rules at the same time. Its ability to cope with the exceptions was even worse. Our best result was 95.73% of correctly transcribed words in the set of 10 thousand words while our rule-based system could correctly transcribe approximately 99.98% of words in the same set.

The neural network could be a good tool for the phonetic transcription if a set of examples with the correct solutions that are governed by only a few rules was available. If the network succeeded to learn all the examples, it could transcribe well all the other words that have the same rules of phonetic transcription. The advantage of this solution would be no need to compile any set of explicit rules.

The neural network is not a good tool in the real-life situations of phonetic transcription where a lot of rules and exceptions exist. Its most serious drawback is the nature of the back-propagation algorithm that usually gets stuck in some local minimum in the error surface that prevents it to find the global minimum – the optimal solution, which is finding of all the rules and exceptions. The languages of a less phonographic nature than the Czech language would also have problems with their representation in the network. Another disadvantage of neural networks is the fact that they cannot be used as tools for finding explicit rules that can be used in the rule-based systems. The rules learned by neural networks are implicit. They are a result of the interaction of all the weights that connect their neurons and take part in the computations transforming the inputs into the outputs.

6.3.6. The Use of Genetic Algorithms for Finding the New Rules for Phonetic Transcription

Having learned that a neural network cannot offer a satisfactory solution to our task of phonetic transcription and at the same time knowing that our existing rule-based system makes so many mistakes that laborious manual corrections of its output are inevitable, we have approached the task of automated finding of new rules for our rule-based system. The method that we have successfully tried is based on the sub-field of genetic algorithms called grammatical evolution. Our solution is published in [68]. Grammatical evolution is for the Czech readers explained in [69] (pp. 148 – 152). The outline of genetic algorithms is e.g. in [3] (pp. 619 – 621).

Genetic algorithms generate a great number of possible solutions and select the most promising of them to become the parents of the new solutions, which are similar but not the same as their parents. Sometimes quite different solutions are generated to explore the yet unexamined part of the space of possible solutions. The solutions are individuals that breed their children with mutations. The quality of the solutions is the analogy of the quality of living conditions. Only the individuals that live in the good living conditions can breed and their children can possibly find even better living conditions. The other individuals die.

The individuals in our problem are rules for phonetic transcription in the form (9). These rules can be modified in the process of breeding with the exception of the strings *A* and *B*. The strings *A* and *B* are the same for the entire population of individuals. In this way we can for example find new rules for the transcription of words containing the sub-string *ni* that should be transcribed as [*ni*] (see Chapter 6.3.4). The strings around the string *A* are automatically generated and each rule that originates in this way is individually tested as a part of a fixed list of the basic rules. If the tested rule happens to successfully transcribe the words that were transcribed incorrectly without this rule, and at the same time this rule does not incorrectly transcribe the words that were transcribed correctly without this rule, it is selected either for being the parent of the next generation of individuals or selected as the new rule for our rule-based system when the grammatical evolution is terminated after a certain number of generations. In the course of grammatical evolution also some individuals that spoil transcriptions of some words are selected to become parents.

In this way we have successfully obtained new rules that transcribe words that contain sub-strings *di*, *ti*, and *ni* that should be transcribed as [*di*], [*ti*], and [*ni*] respectively and not as [*ďi*], [*ťi*], and [*ňi*] as the basic rules say. This class of words contains the largest number of exceptions in the Czech pronunciation. The same method can be applied to finding the arbitrary other rules as well. The data needed for the derivation of rules (the training set) are in the form of the set of words that contain a certain string *A* with correct transcriptions that contain a certain string *B*. The derived rules can be usually successfully applied also to some other words not present in the training set.

6.4. Language Model

6.4.1. The Purpose of the Language Model

A language model represents knowledge about language. This knowledge is usually in the form that tells what words are appropriate for a certain context. People can acquire such knowledge in the course of all their verbal communication, and they use this knowledge every time they hear the speech in the language they know, because many words are uttered unclearly. Machines that should recognize continuous speech must also use such knowledge.

6.4.2. *N*-gram Language Model

The *n*-gram LM is used in the majority of continuous speech recognizers. It is also the kind of LM used in our lab for CSR. The *n*-gram LM is a table of conditional probabilities of a word on condition that a certain succession of $n - 1$ words has preceded it in the speech. The probabilities are computed from a large training corpus. Thanks to this fact the *n*-gram LMs are also called statistical LMs. Each conditional probability *P* is expressed by formula (10).

$$P \text{ of the } m\text{-th word in the corpus} = P(w_m | w_{m-n+1}, \ldots, w_{m-1}) \qquad (10)$$

The probability (10) is called "*n*-gram". In terms of counts of word-successions in the corpus the same conditional probability is computed according to formula (11).

$$P \text{ of the } m\text{-th word in the corpus} = \frac{C(w_{m-n+1}, \ldots, w_m)}{C(w_{m-n+1}, \ldots, w_{m-1})} \qquad (11)$$

According to the value of *n* the *n*-gram LM is called zerogram ($n = 0$, all words have the same probability, no need to count them in the corpus, in fact it is no LM), unigram ($n = 1$, all probabilities are relative frequencies of words in the corpus), bigram ($n = 2$), trigram ($n = 3$), quadrigram ($n = 4$). *N*-gram LMs of higher orders are usually not used.

N-gram LM is computed only for the words that are selected for the recognizer's vocabulary. When the vocabulary has *m* words, the number of probabilities in the *n*-gram LM is m^n. The most of commercial recognizers are capable of working with a 60-thousand-word vocabulary and a trigram LM. This is suitable for the English language and some other languages whose words do not have many inflections. The number of probabilities of the resulting LM is $60{,}000^3 = 216{,}000{,}000{,}000{,}000$. In the case of the Czech language the number of words in the vocabulary must be several hundreds of thousands. The current largest vocabulary used for CSR in our lab has 312 thousand words. A trigram LM of such a large vocabulary would have more values than the contemporary hardware can operate with. For this reason we use a bigram LM. The number of probabilities in our LM is $312{,}000^2 = 97{,}344{,}000{,}000$.

6.4.3. Bigram Language Model

The bigram LM estimates that a sentence *I have new shoes.* is more probable than a sentence *I has new shoes.* The reason for this is the fact that in a large training corpus of grammatical English sentences the word *I* is followed by the word *have* more times than by the word *has*. It means that the conditional probability of the word *have* on condition that the word *I* precedes it is higher than the conditional probability of the word *has* on the same condition. Every human language has many such examples.

Our bigram LM is incorporated into our recognizer in the way that is shown in equations (1) and (2) in Chapter 3.2. Equation (12) brings the term representing the LM in these two equations in concordance with equation (10).

$$g(w_{n-1}, w_n) = P(w_m = w_n | w_{m-1} = w_{n-1}) \qquad (12)$$

The index n in equations (1), (2), and (12) does not express the order of the n-gram LM as it did in equations (10) and (11). In equations (1), (2), and (12) it stands for the index of the word in the recognized sentence.

6.4.4. Advantages of the *N*-gram Language Models

1. *N*-gram LMs are flexible enough to allow the recognition of arbitrary utterances.
2. The algorithm of computing of the n-gram statistics is language independent.
3. The algorithm of computing and using of the n-gram LM is relatively simple, especially in the case of very small vocabularies.
4. *N*-gram LMs are left-to-right. It means that they predict the future from the past. For this reason they can be well integrated with the acoustic models based on HMMs that are also left-to-right.
5. *N*-gram LMs can be easily focused on a certain topic domain. The n-grams coming from the in-domain part of the training corpus can get some higher weight than the other n-grams and the two resulting groups of n-grams can be easily merged. The results of this technique are often surprisingly good. See e.g. [70] or [20].
6. *N*-gram LMs describing the probabilities of wordforms can be easily combined with n-gram LMs describing the probabilities of grammatical categories of wordforms. A report about SpeechLab's initial experiments with the resulting class-based LMs can be seen in [71].

6.4.5. Disadvantages and Challenges of the *N*-gram Language Models

1. *N*-gram LMs consist of a very large number of parameters equal, as stated in Chapter 6.4.2, to m^n where m is the size of the vocabulary, and n is the order of the n-gram LM. The vocabulary size m must be usually very large which means that n must be usually smaller than 4.
2. Practically usable n-gram LMs can describe only local dependencies of words, but the real dependencies in the languages go beyond the successions of two or three words. For example, the trigram LM cannot find a syntactic error in a sentence *A man over there have a gun*.
3. *N*-gram LMs should contain probabilities of all possible word-pairs or word-triplets. But no training corpus is large enough to contain every possible succession of two or three

words. Chapter 7.1.2 informs that a 2.6-GB training corpus of plain text contains 60,228,569 different word-pairs composed of words in a 312k vocabulary. The number of word-pairs whose probability must be estimated is in fact $312{,}000^2$. This means that our corpus contains only 0.06% of all possible word-pairs. Some of the word-pairs that are missing in the corpus should really have almost zero probability. Some other missing word-pairs are admissible for the grammar of the given language. The art of language modeling consists in distinguishing the one group of missing word-pairs from the other and assigning appropriate probabilities to the admissible word-pairs. This task is solved by the so-called LM smoothing described in Chapter 6.4.6.

4. *N*-gram LMs must be estimated from substantially large training corpora. Preparation of such corpora is very laborious (see Chapter 6.1.2), and counting of the frequencies of word-successions must be done in some efficient way (see Chapter 7.1.2).
5. Bigram LMs for vocabularies exceeding approximately 10 thousand words must be represented in a way that enables both their fitting into the operating memory of the contemporary PCs and an efficient retrieval of information they contain in the course of speech recognition. Our solution of this problem is indicated in Chapter 3.2.

6.4.6. Smoothing

Smoothing gives the solution to the zero-frequency problem also called “data shortage” or “data sparseness”. We want to make a rule for every possible combination of circumstances that can happen. The rules can be inferred from our training data, but we do not observe each combination there. So, we must develop the rules for the combinations missing in the training data in some other way than directly inferring them from the data. We can make use of the similarities of the data.

In the case of the language modeling, we can utilize the fact that wordforms belong to grammatical categories. We can infer the rules governing the successions of the grammatical categories from our training corpus. If we knew the probability of grammatical category for every wordform, we could assign a bigram probability to every word-pair. The result is the so-called “class-based language model” previously mentioned in Chapter 4 and in point 6 in Chapter 6.4.4. Class-based LMs are not a subject of this thesis. The LM smoothing studied in this thesis is less sophisticated, and we will justify why it is advantageous for us to have it in this way.

All smoothed *n*-gram LMs are derived from the so-called **Maximum Likelihood Estimate** language model (MLE LM). This LM is computed from the training corpus according to formula (11) in Chapter 6.4.2. The notion of MLE is also in [6] (p. 200).

The simplest method of assigning non zero probabilities to all unseen *n*-grams in the training corpus is **smoothing by adding 1** (also called the **Laplace’s law**) and its generalization called **smoothing by adding less than 1** (also called the **Lidstone’s law**) described by formula (13).

$$P(w_m|w_{m-n+1},\ldots, w_{m-1}) = \frac{C(w_{m-n+1},\ldots,w_m)+a}{C(w_{m-n+1},\ldots,w_{m-1})+aV} \tag{13}$$

Parameter a in formula (13) is some number larger than 0 and equal or less than 1.
Parameter V is the size of the recognizer’s vocabulary.

Each method of LM smoothing transfers some probability mass from the *n*-grams that are present in the training corpus to the unseen *n*-grams. The result of smoothing by adding something between zero and one gives either too much probability mass to unseen word-successions while unacceptably reducing the probabilities of very frequent word-successions, or (when the parameter *a* is small enough) the unseen *n*-grams get probabilities that do not match empirical distributions at low frequencies.

The smoothing method of our choice is **Witten-Bell discounting**. It was firstly published as Method C in [72]. The probabilities in a bigram LM smoothed by this method have the values computed according to formulae (14) and (15)

$$P(w_2|w_1) = \frac{C(w_1, w_2)}{C(w_1) + T(w_1)} \text{ if } C(w_1, w_2) > 0 \tag{14}$$

$$P(w_2|w_1) = \frac{T(w_1)}{(V - T(w_1)) \cdot (C(w_1) + T(w_1))} \text{ if } C(w_1, w_2) = 0 \tag{15}$$

where $T(w_1)$ is the number of wordform types that follow wordform w_1 in the training corpus.

The idea of this smoothing is the following: The conditional probability of wordform w_2 on condition that wordform w_1 precedes it is directly in proportion to the number of wordform types that were already observed in the training corpus to follow the wordform w_1. Add-one smoothing and Witten-Bell discounting are in detail explained in [6] (pp. 206 – 214). In the following paragraphs we give some details about Witten-Bell discounting that are not a part of its standard description.

According to the theory of probability, the sum of conditional probabilities on the same condition must be equal to one. In the case of language modeling it means that the sum of conditional probabilities of words on condition that a certain sequence of words precedes them is equal to one. Identity (16) is important for understanding further formulas in this chapter. It tells us that the sum of counts of all word sequences that end in the same wordform w_m is equal to the count of the shorter sequence in which the last wordform w_m is missing. In the case of the bigram LM this identity is in the form (17).

$$\sum_{w_m} C(w_{m-n+1}, \ldots, w_m) = C(w_{m-n+1}, \ldots, w_{m-1}) \tag{16}$$

$$\sum_{w_2} C(w_1, w_2) = C(w_1) \tag{17}$$

Formula (16) and its special case (17) can help to prove that the MLE, add-one, and Witten-Bell LM obey the abovementioned law of conditional probability. For example, the fact that a sum of bigrams ending with the same wordform in the Witten-Bell LM is equal to one can be proved by formulae (18) and (19). The sum of probabilities of bigrams beginning with the same wordform w_1 that are present in the training corpus (see formula (14)) is expressed by identity (18).

$$\sum_{w_2} P(w_2|w_1) = \sum_{w_2} \frac{C(w_1, w_2)}{C(w_1) + T(w_1)} = \frac{C(w_1)}{C(w_1) + T(w_1)} \tag{18}$$

The sum of probabilities of bigrams beginning with the same wordform w_1 that are not present in the training corpus (see formula (15)) is expressed by identity (19).

$$\begin{aligned}\sum_{w_2} P(w_2|w_1) &= \sum_{w_2} \frac{T(w_1)}{(V-T(w_1))\cdot(C(w_1)+T(w_1))} = \\ &= \frac{(V-T(w_1))\cdot T(w_1)}{(V-T(w_1))\cdot(C(w_1)+T(w_1))} = \frac{T(w_1)}{C(w_1)+T(w_1)}\end{aligned} \tag{19}$$

It is easy to see that the sum of the right sides of formulas (18) and (19) is equal to one.

When using the Witten-Bell discounting in the real applications, we must make rules for every possible situation to prevent run-time errors that could terminate several hours long processing of the list of word-pairs whose purpose is the computation of the smoothed LM.

The only case that can end up in a run-time error is the situation when some word in the recognizer's vocabulary is missing in the training corpus. This can happen when we have compiled a vocabulary for some special domain for which we do not have any training corpus available. The easiest way of the solution to this problem is to give bigrams that begin with the wordform w_1 missing in the training corpus a uniform distribution of probability (20).

$$P(w_2|w_1) = \frac{1}{V} \text{ if } C(w_1) = 0 \tag{20}$$

When we look to the formula (15), we can conclude that another situation that can end up in a division-by-zero error is when $T(w_1) = V$. Then we realize that this situation can't happen because in this situation the formula (15) is never used. But when this situation is projected to the equations (14) and (18), we can see that the sum of probabilities of bigrams beginning with the same wordform w_1 is less than one.

But there is yet another situation in which the standard formulas for Witten-Bell discounting do not return a satisfying output. It is the situation when $T(w_1) > V/2$. Formulae (14) and (15) have in common term (21).

$$\frac{1}{C(w_1)+T(w_1)} \tag{21}$$

The rest of formula (15) is higher than one.

$$\frac{T(w_1)}{V-T(w_1)} > 1 \text{ if } T(w_1) > \frac{V}{2} \tag{22}$$

The rest of formula (14) can be equal to one or higher than one (it is the count of the word-pair w_1, w_2 in the training corpus), which means that sometimes the Witten-Bell discounting gives higher probabilities to the word-pairs unseen in the training corpus than to the word-pairs that have the same first word and were seen in the training corpus, because

$$\frac{T(w_1)}{V-T(w_1)} \text{ is surely higher than 1, but } C(w_1, w_2) \text{ can be equal to 1.} \tag{23}$$

We can solve this situation by applying for example the add-one smoothing (24) to the affected lines of the bigram LM. So far, we have been using this approach, see [70].

$$P(w_2|w_1) = \frac{C(w_1, w_2) + 1}{C(w_1) + V} \quad \text{if } 2T(w_1) > V \tag{24}$$

Or we can design new formulae (25) and (26) to avoid the drawbacks of add-one smoothing.

$$P(w_2|w_1) = \frac{C(w_1, w_2) \cdot (C(w_1) + 2T(w_1) - V)}{C(w_1) \cdot (C(w_1) + T(w_1))} \quad \text{if } 2T(w_1) > V \text{ and } C(w_1, w_2) > 0 \tag{25}$$

$$P(w_2|w_1) = \frac{1}{C(w_1) + T(w_1)} \quad \text{if } 2T(w_1) > V \text{ and } C(w_1, w_2) = 0 \tag{26}$$

With the help of identity (17) we can prove that formulae (25) and (26) produce conditional probabilities that are in compliance with the probability theory:

$$\sum_{w_2} P(w_2|w_1) = \frac{C(w_1) \cdot (C(w_1) + 2T(w_1) - V)}{C(w_1) \cdot (C(w_1) + T(w_1))} = \frac{C(w_1) + 2T(w_1) - V}{C(w_1) + T(w_1)} \quad \text{if } C(w_1, w_2) > 0 \tag{27}$$

$$\sum_{w_2} P(w_2|w_1) = \frac{V - T(w_1)}{C(w_1) + T(w_1)} \quad \text{if } C(w_1, w_2) = 0 \tag{28}$$

The sum of the right sides of formulae (27) and (28) is again equal to one. Moreover, the formula (25) reasonably solves the situation when $T(w_1) = V$.

$$P(w_2|w_1) = \frac{C(w_1, w_2) \cdot (C(w_1) + 2V - V)}{C(w_1) \cdot (C(w_1) + V)} = \frac{C(w_1, w_2)}{C(w_1)} \tag{29}$$

If $T(w_1) = V$, it is no longer any reason for smoothing, and the appropriate LM is MLE.

In Table 21 in Chapter 7.4.3 we present the conditional cross perplexity of LM against two test corpora. The LM is smoothed by Witten-Bell discounting in its standard form (14), (15), and (20), Witten-Bell discounting combined with add-one smoothing (24), and the improved Witten-Bell discounting modified by formulae (25) and (26).

Many other kinds of LMs are successfully used in the research community dealing with ASR. Some of these LMs can provide a more precise description of the given language than e.g. Witten-Bell discounting. We do not plan to use these better LMs in the near future, because we would have problems with their implementation. Let us explain this on the example of the popular linear interpolation smoothing.

The trigram LM smoothed by the **linear interpolation smoothing** has probabilities computed according to formulae (30) and (31).

$$P(w_3|w_1, w_2) = \lambda_3 P(w_3|w_1, w_2) + \lambda_2 P(w_3|w_2) + \lambda_1 P(w_3) + \lambda_0 / V \tag{30}$$

$$\sum_{i=0}^{3} \lambda_i = 1 \tag{31}$$

The probabilities in this LM are a result of a weighted sum of conditional probabilities found in the training corpus. This kind of smoothing solves the zero-frequency problem using the following rules: If a particular trigram $P(w_3|w_1, w_2)$ is not present in the training corpus, let it be estimated with the help of a bigram $P(w_3|w_2)$ with a shorter one-word history. If such bigram is also missing in the corpus, let it be estimated with the help of the frequency of a single word $P(w_3)$. And when even the single word w_3 is missing in the corpus, let is be substituted by a uniform unigram probability (the zerogram LM) $1/V$. This method was introduced in [73] as **deleted interpolation**. The weights λ in formula (30) can be optimized in such a way that the perplexity (explained in Chapter 6.4.11) of the resulting LM on the training corpus is minimized. Besides the accuracy of recognition, the perplexity is the second most important parameter of the LM, because it can be observed that ASR systems with LMs with a lower perplexity on the recognized text have usually a higher accuracy of recognition. In other words, in language modeling we should always strive to construct LMs with the lowest possible perplexity. So, the linear interpolation smoothing looks like an ideal kind of LM. We do not use this smart LM for the following reasons:

1. The LM smoothed by linear interpolation is not normalized. It means that the sum of conditional probabilities of words on condition that a certain sequence of words precedes them is not equal to one. The formula (31) is not enough to ensure the normalization. The truth is that in most applications this LM serves well even without being normalized, because the only thing that matters is its minimized perplexity.
2. The LM smoothed by linear interpolation contains too many distinct values. In Chapter 3.2 we explain the necessity of compressing our LM based on a 312k vocabulary. Our compression algorithm takes advantage of the fact that all the bigrams beginning with the same word w_1 and having the same count $C(w_1, w_2)$ have the same probability in the MLE, add-one, and Witten-Bell LMs respectively. Formula (30) modified for the case of the bigram LM conditions its result not only on $C(w_1, w_2)$ and $C(w_1)$ (the probability $P(w_2|w_1)$) but also on $C(w_2)$ (the probability $P(w_2)$). Moreover, the linear interpolation LM usually outperforms the other LMs only if it is computed using the so-called "bucketed smoothing", which consists in optimizing several sets of λ parameters for distinct histories of n-grams. This improvement also extends the variety of probabilities in the resulting n-gram table. The linear interpolation LM is usually supposed to be kept in memory as a table of easy-to-compress MLE probabilities $P(w_3|w_1, w_2)$, $P(w_3|w_2)$, $P(w_3)$, and $1/V$, together with several sets of the λ weights. The smoothed probabilities are computed in the course of recognition. This means that for a given history of words w_1, w_2 every possible word w_3 must be tested. Our recognizer makes use of the fact that the values of its LM are not only compressed but also sorted, so that only the most probable bigrams starting with a certain word are tested. If we had to implement the LM smoothed by linear interpolation, we should quantize its resulting probabilities, so that they become less variable, and then sort each row of our LM.
3. The linear interpolation of MLE LMs is a relatively computationally intensive algorithm. The training corpus must be divided into the so-called "training" part and "held-out" part. The MLE probabilities are computed from the training part of the corpus. Then the linear interpolation algorithm in several iterations updates the λ weights, so that the perplexity of the resulting LM on the held-out corpus is minimal. In each iteration a long list of word-pairs existing in the training corpus is processed. In our case it would be about 60 million word-pairs. In contrast, the computation of the Witten-Bell LM takes only two passes through the list of word-pairs.

6.4.7. Implementation of the Bigram Language Model

We must work with vocabularies containing more than 300k words. The bigram LM based on a several-hundred-thousand vocabulary is so large that it must be compressed. In Chapter 3.2 we reveal our representation of the bigram LM in the computer memory. In point 2 in Chapter 6.4.6 we describe how dependant our algorithm is for LM compression and a retrieval of information out of it on the method of LM smoothing. Only the LMs like MLE, add-one, Witten-Bell, and the likes of them are acceptable for us.

6.4.8. Appropriateness of the Bigram Language Model for the Czech Language

The Czech language has many properties that impair the help of the standard *n*-gram LMs based on wordforms in ASR. Let us depict the most important difficulties.

The Czech language has a very rich vocabulary of wordforms. This leads to the need of working with very large LMs (occupying too much of computer memory space) that can be maximally of the second order (the bigram LMs).

The Czech language is very inflective. It is the reason for its rich vocabulary. At the same time, the Czech language is governed by many rules of gender, number, and case agreement within the sentences. A considerable number of word-pairs should not appear in a grammatical Czech sentence. This results in very sparse MLE LMs that must be smoothed somehow, but smoothing that does not take into account the information about the Czech grammatical agreement gives too much probability to the bigrams that should retain a zero probability. This problem is depicted in Table 4. The Czech part of the table is both larger and sparser than the English part.

Table 4. The comparison of the corresponding parts of the Czech and the English bigram LM containing all possible pronoun subjects and a verb *jít* (*to go*). The gray fields should contain zero probabilities.

	jdu	*jdeš*	*jde*	*jdeme*	*jdete*	*jdou*			*go*	*goes*
já		▨	▨	▨	▨	▨		*I*		▨
ty								*you*		▨
on	▨	▨		▨	▨	▨		*he*	▨	
my	▨	▨	▨		▨	▨		*we*		▨
vy	▨	▨	▨	▨		▨		*you*		▨
oni	▨	▨	▨	▨	▨			*they*		▨

A side effect of the rules of grammatical agreement is a relatively free word order in the Czech sentence. The grammatical agreement carries the information about the relations of subjects and objects, so there is no need to order the words in Czech sentences. This reduces the LM sparseness, but at the same time the *n*-gram probabilities become less distinct in predicting the correct words from the previous words. The example of the variability with which a single Czech sentence can be formulated is given in Table 5.

Table 5. The free word order in the Czech sentence. The ordering of words often expresses the stress on a particular part of the sentence.

I can't repair that engine myself. can be expressed in the Czech language as:
Nemohu opravit ten motor sám. *Nemohu sám opravit ten motor.* *Sám nemohu opravit ten motor.* *Sám nemohu ten motor opravit.* *Sám ten motor nemohu opravit.* *Ten motor nemohu opravit sám.*
And still many other variants are possible.

Thanks to the fact that the Czech language is partly a product of the linguists living 200 – 300 years ago (Josef Dobrovský, Josef Jungmann, and František Palacký to name a few) who were compiling the dictionaries of literary (standard) Czech according to some old Czech literature whilst the folk was speaking using Germanisms and colloquialisms, there is still a distinct gap between the spoken and written form in the Czech language. It would be for example difficult to compile a corpus of the Czech colloquial spontaneous speech, because the most of the written Czech language is in the literary form. The language in which the Czech broadcast news are presented is however mostly standard.

All these properties suggest that a simple statistical approach is not sufficient for ASR of the Czech language. Nevertheless, this thesis tries to explore the potential of statistical methods in this task.

6.4.9. Collocations in the Bigram Language Model

Treating some word-successions as single words influences the quality of the *n*-gram LM. Such word-successions are called collocations and they are described in Chapter 6.2.10. Their effect on the bigram LM was published in [51]. The description of this problem given here is more detailed.

Table 6 shows a part of the bigram Maximum Likelihood Estimate LM with selected words that are not joined into collocations. The gray fields carry zero conditional probability according to some hypothetical data.

Table 6. The part of the bigram MLE LM without words joint into collocations. The fact that the words *Burkina* and *is* have more than one Czech equivalent is an example of Czech inflection.

	Burkina (*Burkina*)	*Burkině* (*Burkina*)	*Faso* (*Faso*)	*je* (*is*)	*byla* (*is*)	*s* (*with*)	*v* (*in*)
Burkina	▨	▨		▨	▨	▨	▨
Burkině	▨	▨		▨	▨	▨	▨
Faso	▨	▨	▨				
je		▨	▨	▨	▨		
byla		▨	▨	▨	▨		
s		▨	▨	▨	▨	▨	▨
v	▨		▨	▨	▨	▨	▨

Table 7 shows the same part of the LM as Table 6 but the words Burkina and Faso are joined into a collocation.

Table 7. The part of the bigram MLE LM with some words joint into collocations. The indications of zero probability (the gray fields) correspond with Table 6.

	*Burkina_* *Faso*	*Burkině_* *Faso*	*je* (*is*)	*byla* (*is*)	*s* (*with*)	*v* (*in*)
Burkina_Faso	▨	▨				▨
Burkině_Faso	▨	▨			▨	
je		▨	▨	▨		
byla		▨	▨	▨		
s		▨	▨	▨	▨	▨
v	▨		▨	▨	▨	▨

Table 6 and Table 7 demonstrate that treating collocations as single words removes from the bigram LM the sparse lines dedicated to collocations. Otherwise such lines (in our example they are the first two lines in Table 6) would have been given some inappropriate probabilities in the course of smoothing.

See Chapter 7.4.2 for experimental results showing the importance of joining words into collocations for CSR.

6.4.10. Compound Words in the Bigram Language Model

Decomposition of some wordforms into several parts affects the quality of the *n*-gram LM. We were experimenting only with the so-called compound words characterized in Chapter 6.2.11. The effect of their decomposition on the bigram LM was shown in [51]. Here we want to present this problem in a more comprehensible way.

Table 8 shows a part of the bigram MLE LM with some selected words. The gray fields carry zero conditional probability according to some hypothetical data.

Table 8. The part of the bigram MLE LM with whole words

	jít (*to go*)	*nejít* (*not to go*)	*vidět* (*to see*)	*nevidět* (*not to see*)	*větší* (*bigger*)	*největší* (*the biggest*)	*menší* (*smaller*)	*nejmenší* (*the smallest*)
jít	▨	▨	▨	▨	▨			▨
nejít	▨	▨	▨	▨	▨	▨		
vidět	▨	▨	▨	▨	▨	▨		
nevidět	▨	▨	▨	▨				▨
větší			▨	▨	▨	▨	▨	▨
největší	▨			▨	▨	▨	▨	▨
menší	▨	▨			▨	▨	▨	▨
nejmenší				▨	▨	▨	▨	▨

Table 9 shows the same part of the LM as Table 8 but the words that contained prefixes *ne* and *nej* are now decomposed.

Table 9. The part of the bigram MLE LM with decomposed words. The placement of zero probability (the gray fields) is derived from Table 8.

	ne (*not*)	*nej* (*the *est*)	*jít* (*to go*)	*vidět* (*to see*)	*větší* (*bigger*)	*menší* (*smaller*)
ne	▨	▨			▨	▨
nej	▨	▨	▨	▨		
jít	▨		▨	▨	▨	
vidět	▨		▨	▨	▨	
větší		▨			▨	▨
menší		▨			▨	▨

Table 8 and Table 9 demonstrate how the decomposition of words reduces the vocabulary and the sparseness of the LM. The percentage of non-zero (white) fields in Table 9 is bigger than in Table 8.

See Chapter 7.4.3 for experimental results showing the importance of decomposition of words into their parts for CSR of the Czech language.

6.4.11. Evaluation

The most important criterion of the quality of LM is its influence on the accuracy of recognition. The results of this kind are reported in Chapter 7.4.

However, it is important to test the quality of the LM using only the text information about the test corpus. The result of this test is independent of the acoustic quality of the recordings transcriptions of which the test corpus consists. The quality of the LM itself and the measure of how the LM matches the test corpus is usually in the form of **perplexity** defined by the information theory.

Perplexity in this thesis is marked by letter G, and it is derived from the so-called **entropy** which is marked by letter H. The information theory distinguishes several kinds of entropies. The relation between perplexity and entropy of all kinds is expressed by formula (32)

$$G = 2^{H} \tag{32}$$

The basic formula for entropy is (33)

$$H_1 = -\sum_{x\in\Omega} p(x)\cdot\log_2 p(x) \tag{33}$$

where $p(x)$ is the probability of the event x. The set of all possible events x is Ω, also called the alphabet. The result of formula (33) is in bits. It is the average information content of the various events x weighted by the probabilities of the events.

The result of formula (32) is the average number of events from which we must choose one when predicting the random value of x. When the probability of all possible values of x is the same, then the entropy is maximal and perplexity is equal to the number of all possible values in the alphabet of x.

Perplexity of the LM is derived from the so-called **conditional entropy** defined by formula (34)

$$H_2 = -\sum_{x\in\Omega}\sum_{y\in\Omega} q(x,y)\cdot\log_2 q(y|x) \tag{34}$$

where $q(x, y)$ is the probability of a word-pair whose first word is x and the second word is y. This probability is equal to the count of the word-pair x, y divided by the number of all tokens in the training corpus. Ω is the set of all words in the vocabulary. $q(y|x)$ is the conditional probability of the word y on condition that the word x precedes it. In terms of word-counts $q(y|x)$ is computed according to equation (11), i.e. $C(x, y)$ divided by $C(x)$. $q(y|x)$ may also be a result of bigram LM smoothing.

The closeness of the LM to the test corpus is expressed by the perplexity derived from the so-called **conditional cross entropy** described by formula (35)

$$H_3 = -\sum_{x\in\Omega}\sum_{y\in\Omega} p(x,y)\cdot\log_2 q(y|x) \tag{35}$$

where $p(x, y)$ is the probability of a word-pair x, y in the test corpus, and $q(y|x)$ is the conditional probability of the word y on condition that the word x precedes it in the training corpus. The table of all $q(y|x)$ is the bigram LM computed from the training corpus. Words x and y belong to the same alphabet which is the set of all words in the recognizer's vocabulary.

Because of the fact that logarithm of zero is undefined, it is desirable to set all probabilities in the LM to some values above zero, hence another reason for the LM smoothing.

In the case when the test corpus contains some out-of-vocabulary (OOV) words, two possible methods of computation of conditional cross entropy exist. The first method is to ignore the OOV words. The second method is to map all the OOV words to the same special word "OOV", and this word is added to the vocabulary according to which the LM is computed. The results in this thesis are computed using the first method.

Table 10. Computation of conditional cross entropy

Language Model - Bigram Counts					
	a	*b*	*c*	*d*	**Sum**
a	10	20	10	10	50
b	10	10	10	10	40
c	10	10	10	10	40
d	10	10	10	10	40
Sum = Number of Tokens					170
Language Model - Bigram Conditional Probabilities $q(y \mid x)$					
	a	*b*	*c*	*d*	**Sum**
a	0.2	0.4	0.2	0.2	1
b	0.25	0.25	0.25	0.25	1
c	0.25	0.25	0.25	0.25	1
d	0.25	0.25	0.25	0.25	1
Sum = Vocabulary Size					4
Language Model - Entropy Summands					
	a	*b*	*c*	*d*	**Sum**
a	23.22	26.44	23.22	23.22	96.096
b	20	20	20	20	80
c	20	20	20	20	80
d	20	20	20	20	80
Sum					336.096
Language Model Entropy = **= Sum / Number of Tokens**					1.97704

Test Data - Bigram Counts					
	a	*b*	*c*	*d*	**Sum**
a	1	2	1	1	5
b	1	1	1	1	4
c	1	1	1	1	4
d	1	1	1	1	4
Sum = Number of Tokens					17
Test Data - Bigram Conditional Probabilities $p(y \mid x)$					
	a	*b*	*c*	*d*	**Sum**
a	0.2	0.4	0.2	0.2	1
b	0.25	0.25	0.25	0.25	1
c	0.25	0.25	0.25	0.25	1
d	0.25	0.25	0.25	0.25	1
Sum = Vocabulary Size					4
Test Data - Cross Entropy Summands					
	a	*b*	*c*	*d*	**Sum**
a	2.322	2.644	2.322	2.322	9.6096
b	2	2	2	2	8
c	2	2	2	2	8
d	2	2	2	2	8
Sum					33.6096
Conditional Cross Entropy = **= Sum / Number of Tokens**					1.97704

Table 10 shows the example of computation of the conditional cross entropy. The vocabulary of this task is a set of words *a*, *b*, *c*, *d*. The circled number is the count of all word-pairs *a*, *b* found in the training corpus. This number might have also been the result of LM smoothing. Examples of computations of intermediate results follow:

Number 26.44 = – 20 · $\log_2 0.4$ where both numbers have been taken from the LM.

Language model entropy 1.97704 = 336.096 / 170.

This is the example of the use of formula (34).

Number 2.644 = – 2 · $\log_2 0.4$ where 0.4 is taken from the LM.

Conditional cross entropy 1.97704 = 33.6096 / 17.

This is the example of the use of formula (35).

In the example in Table 10 the LM conditional entropy is equal to the conditional cross entropy. It can be seen that this happens only when the probabilities of the training data are the same as the probabilities of the test data.

In real large corpora these two entropies can never have exactly the same values. But we must know what LM has its probability distribution closest to the test data, because such LM is the best. In further text we prove that the best LM has the lowest conditional cross entropy.

In Table 11 we show the resulting values of LM conditional entropy and conditional cross entropy computed from the data given in Table 10. In these data we were changing the count of the word-pair a, b in the LM while the rest of the input values stayed unchanged.

Table 11. Sensitivity analysis of entropy computed in Table 10

(a, b) Count	**LM Conditional Entropy**	**Conditional Cross Entropy**
5	1.988757563	2.060986742
10	2	2
20	1.977037675	1.977037675
30	1.930827083	1.985581618

It can be seen in the column for conditional cross entropy in Table 11 that this entropy is really minimal when it is equal to the LM conditional entropy, and that it increases as the probability distribution of the LM differs more from the probability distribution of the test data.

It can be seen in the column for LM conditional entropy in Table 11 that this entropy is maximal when the probabilities of all word-pairs in the LM are the same. In this case the LM perplexity (2^2) is equal to the number of words in the vocabulary of the LM. This entropy decreases as the probability distribution of the LM becomes more biased.

The formal proof that the conditional cross entropy is the lowest for the LMs with the probability distributions closest to the test data is the following:

Formula (35) for conditional cross entropy can also have a form

$$H_3 = -\sum_{x\in\Omega} \frac{C(x)}{\#TestTokens} \sum_{y\in\Omega} p(y|x) \cdot \log_2 q(y|x) \tag{36}$$

where *#TestTokens* is the sum of counts of all words in the test corpus that belong to the vocabulary of the LM $q(y|x)$.

The inner sum in formula (36) is in the form of the so-called **cross entropy**

$$H_4 = -\sum_{x\in\Omega} p(x) \cdot \log_2 q(x) \tag{37}$$

where $p(x)$ and $q(x)$ are different probability distributions of the event x.

If we prove that the cross entropy (37) is minimal when $p(x)$ is equal to $q(x)$ for every possible event x, we can apply this law to the conditional cross entropy between the probability distributions of the word-pairs in the LMs and the test corpus.

Because of the fact that in formula (37) the sums of probabilities have identities (38)

$$\sum_{x\in\Omega} p(x) = \sum_{x\in\Omega} q(x) = 1, \tag{38}$$

we can also say that

$$\sum_{x\in\Omega} p(x) \cdot \log_2 q(x) = \overline{\log_2 q(x)} = \text{the average } \log_2 q(x). \tag{39}$$

The cross entropy (37) is a sum of the entropy of the distribution of $p(x)$ and the so-called **Kullback-Leibler distance (relative entropy)** $D(p\|q)$ between $p(x)$ and $q(x)$.

$$H_4 = H_1 + D(p\|q) \tag{40}$$

where

$$D(p\|q) = \sum_{x\in\Omega} p(x) \cdot \log_2 \frac{p(x)}{q(x)}. \tag{41}$$

Indeed:

$$\begin{aligned} H_1 + D(p\|q) &= -\sum_{x\in\Omega} p(x) \cdot \log_2 p(x) + \sum_{x\in\Omega} p(x) \cdot \log_2 \frac{p(x)}{q(x)} = \\ &= -\sum_{x\in\Omega} p(x) \cdot \log_2 p(x) + \sum_{x\in\Omega} p(x) \cdot \log_2 p(x) - \sum_{x\in\Omega} p(x) \cdot \log_2 q(x) = \\ &= -\sum_{x\in\Omega} p(x) \cdot \log_2 q(x) = H_4 \end{aligned} \tag{42}$$

We can be sure that the entropy H_1 (33) is always above or equal to zero. If we prove that $D(p\|q)$ is also always above or equal to zero, we will get the evidence that the cross entropy H_4 is always above or equal to the entropy H_1. The entropy H_1 is equal to H_4 (i.e. minimal) when the probability distribution in the test data is equal to the probability distribution in the LM.

For the proof we compare the average of logarithms with the logarithm of average. If we compare formula (41) with formula (39), we can see that $D(p\|q)$ is the average of logarithms.

$$-D(p\|q) = \sum_{x\in\Omega} p(x) \cdot \log_2 \frac{q(x)}{p(x)}. \tag{43}$$

Now, we must construct an analogical logarithm of average (44):

$$\log_2 \sum_{x\in\Omega} p(x) \cdot \frac{q(x)}{p(x)} = \log_2 \sum_{x\in\Omega} q(x) = \log_2 1 = 0. \tag{44}$$

If we compare equation (41) with equation (43), we can see the difference in the sign and the inverse fraction. This modification was done, so that we could compare the analogical logarithm of average (44) with zero.

So, we have a proof that the logarithm of average of some function is equal to zero. What else should be done is to compare with it the analogical average of logarithms. We can see in Figure 6 that a logarithm of an average is always equal to or higher than an average of logarithms.

Now we can put these facts together. We have seen that the average of logarithms (43) should be always equal to or smaller than the logarithm of the average (44) which is always equal to zero.

$$\begin{gathered} -D(p\|q) \le 0 \\ D(p\|q) \ge 0 \end{gathered} \tag{45}$$

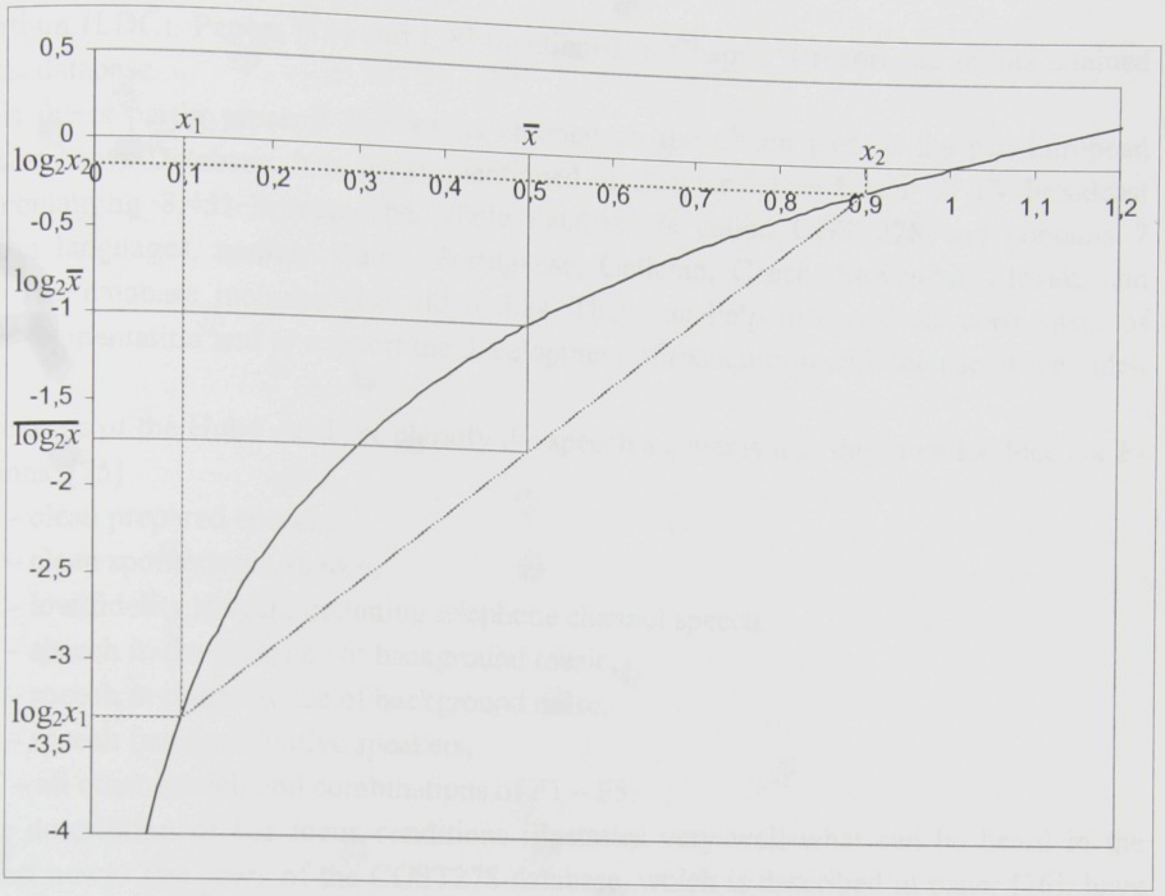


Figure 6. The comparison of the average of logarithms with the logarithm of average

In this way we have proven that the LM with the probability distribution closest to the test data has the lowest conditional cross entropy with the test data.

The literature on this topic is for example [74].

6.5. Test Speech Database

The purpose of test speech databases is to estimate the performance of the recognizer on real data under the real conditions. Before this testing the recognizer was tuned on some training and development databases. The process of tuning of our recognizer is described in the following Chapter 6.6. The estimation of the recognizer's performance is usually done by manually processing the test speech database the same way the recognizer should process it. The result of the recognizer is then compared to the result of manual processing. The recognizer should rewrite the speech into text. The intermediate necessary task of the recognizer is to segment the whole broadcast news show into speech/non-speech parts. The speech parts should also be automatically segmented into so-called speaker turns which are segments with the same speaker and the same acoustic condition, because this enables the adaptation of the recognizer to a certain speaker or acoustic condition. So, besides the transcription of speech also automatic segmentation can be tested on the test speech databases.

The creation of test speech databases is very often a subject of standardization. In this way a single database can be tested by many independent recognizers, and their results can be easily compared. One of the most important databases of that kind is the so-called Hub4 or HUB-4 American Broadcast News text and acoustic corpus collected by the Linguistic Data

Consortium (LDC). Papers [19] and [38] mentioned in Chapter 4 report the results attained with this database.

This thesis partly presents the results attained on the Czech part of the pan-European Broadcast News Database. It is firstly mentioned in Chapter 3.1 as 1 hour of TV broadcast news containing 8,451 words. The whole database is called COST278 and contains 7 European languages, namely Dutch, Portuguese, Galician, Czech, Slovenian, Slovak, and Greek. The database includes also video files. They can help to check the correctness of speaker segmentation and to support the development of recognition with the use of the video part.

The users of the Hub4 database classify the speech segments into the so-called focus or F-conditions: [75]

F0 – clean prepared speech,

F1 – clean spontaneous speech,

F2 – low fidelity speech, including telephone channel speech,

F3 – speech in the presence of background music,

F4 – speech in the presence of background noise,

F5 – speech from non-native speakers,

FX – all other speech and combinations of F1 – F5.

The description of the focus conditions illustrates very well what can be heard in the broadcast news. The users of the COST278 database, which is described in paper [76], have adopted the same classification.

A little test speech database has been created for the purpose of this thesis. Its source is radio broadcast news. The recordings were manually segmented according to grammatical sentences into single files. The file names enable to discern speech without any stumbling, speech with a slip of the tongue, low fidelity speech, and speech in the presence of background music or noise. Another dimension of this description is the names of the speakers and their classification into professional speakers and guests, and into males and females. The source for the reference transcriptions was www.tamtam.cz. The cutting of each broadcast news show into sentences and providing each audio file with the text file containing manually checked reference transcription was done with the use of the SpeechLab's software mentioned in [77]. The resulting database is described in Chapter 7.5.

6.6. Tuning of the Recognizer's Parameters

The recognizer used in all experiments described in this thesis is described in Chapter 3. The following parameters influence the quality of recognition:

1. *Acoustic model* (HMMs in the form of either 16 or 32-mixture monophones),
2. *Language model*,
3. *LM Factor*,
4. *Word Insertion Penalty*,
5. *Prune Threshold*,
6. *Number of Word-End Hypotheses*,
7. *Recognizer's vocabulary*.

The first task after the successful development of our recognizer was to find an optimal combination of all its parameters. This task is very difficult because each parameter influence the effect of the other parameters. It is not usually sufficient to have all parameters fixed and change one of them to find its optimal value. It is necessary to search in the space of several dimensions. But a systematic search in the most promising part of this space would take several thousands of experiments. It is impossible to make so many experiments because each of them lasts several minutes or even hours. The time spent by a single experiment depends on the number of sentences in the development test set. The size of this set should not be too small, because in this case the so-called over-training could occur. The set of parameters is over-trained when its performance on the devtest set is significantly higher than on all other sets. The CSR task is according to our experiences very prone to over-training.

We have used a devtest set of 800 and later 1,600 sentences to tune our parameters. After several initial experiments we decided to search for the right combination of parameters stated above in the space of the *Language model*, *LM Factor*, and the *Word Insertion Penalty*. In the case of the *Acoustic model* just a small number of experiments can prove that 32-mixture monophones can attain a better accuracy than 16-mixture monophones with only a moderate increased time consumption, see results in [11] and [16]. The role of the *Prune Threshold* and the *Number of Word-End Hypotheses* parameters is to prune a search tree of possible words in the recognized sentence. The *Prune Threshold* parameter does this on the HMM state level, and the *Number of Word-End Hypotheses* parameter does this on the word level. Roughly stated, the higher these parameters are, the higher is the accuracy of recognition, but each of them has a certain level of saturation. When the parameter reaches this level, the accuracy rises only negligibly, but the time consumption still rises. The *Recognizer's vocabulary* parameter is very dependant on the test data. In our initial tuning experiments with the devtest set of 800 and 1,600 sentences we have used a fixed closed vocabulary. Later we were experimenting with open vocabularies. In these experiments we have studied the effect of the vocabulary size, collocations, word-decomposition, and multiple phonetic transcriptions.

The results of our first CSR experiments have lead us to develop the following methodology of parameter tuning:

1. Use only 32-mixture monophone HMMs as the *Acoustic model*.
2. Find some reasonable level of the *Prune Threshold* and the *Number of Word-End Hypotheses* parameters. According to our experience these two parameters do not influence the ideal combination of the other parameters too much [11]. The ideal combination of values of these two parameters will be fixed for the subsequent points 3 and 4.
3. For each particular *Language model* find the optimal combination of the *LM Factor* and the *Word Insertion Penalty* parameters. See our experimental results in Table 17 and Table 18 in Chapter 7.4.1 and Figure 10 in Chapter 7.6.
4. Choose the best *Language model* as the one with which the highest accuracy of recognition was attained in point 3. We have chosen the Witten-Bell discounting LM.
5. For the best *Language model* and its optimal combination of the *LM Factor* and the *Word Insertion Penalty* parameters find the saturated level of the *Prune Threshold* and the

Number of Word-End Hypotheses parameters. See our experimental results in Figure 11 and Figure 12 in Chapter 7.6.

The transition from the closed vocabulary of 7,033 words for our 1,600-sentence devtest set used in 2002 to the open vocabulary of 312,000 words that we started to use in 2005 has changed the ideal combination of the parameters found in 2002 substantially. Our large vocabulary can be tested only on a large test set, and such an experiment lasts several hours. So we have done only a few tuning experiments to find out new ideal values of parameters. These values can be seen in Table 24 in Chapter 7.6.

6.7. Evaluation of Recognition

The standard measures of the quality of recognition are based on the differences between the reference transcription and the output of the recognizer. It is possible to use the following formulae [11], [6] (p. 271):

$$Correctness\ [\%] = 100 \cdot \frac{N - D - S}{N} \tag{46}$$

$$Accuracy\ [\%] = 100 \cdot \frac{N - D - S - I}{N} \tag{47}$$

$$Word\ Error\ Rate = WER\ [\%] = 100 \cdot \frac{D + S + I}{N} = 100 - Accuracy \tag{48}$$

N is the total number of words in the correct transcription.

D is the number of word deletions.

S is the number of word substitutions.

I is the number of word insertions.

D, *S*, and *I* are a result of the **minimum edit distance** algorithm described e.g. in [6] (pp. 153 - 156). The result of this algorithm, the sum of the number of deletions, substitutions, and insertions, is sometimes called the **Levenshtein distance**. In one variant of this measure each of the three operations has a cost of 1. In the other variant the operation of deletion and insertion has a cost of 1, and the operation of substitution has a cost of 2, because it is an equivalent of one deletion and one insertion. We use the former variant to compute the accuracy of recognition. In Appendix 1 of this thesis is an example of the output of our recognizer and the minimum edit distance algorithm applied to it.

Another important measure of the quality of recognition is the time consumption. This is expressed as the so-called **real-time factor** *xRT*:

$$xRT = \frac{Duration\ of\ recognition}{Duration\ of\ the\ recognized\ utterance} \tag{49}$$

The speed of recognition measured by formula (49) is important especially for the recognition tasks that must be performed in real-time, e.g. for the purpose of providing the current broadcast TV news show with closed caption. The disadvantage of the real-time factor is its dependency on hardware. It should always be presented together with the information what computer was used for the recognition.

The output of the recognizer may also be the input of the automatic information retrieval system (see e.g. [6] (pp. 631 – 666)). In this case the most important would be the measure of *Correctness*, *Accuracy* or *WER* but only for the certain key words that characterize the content of the document.

The performance of the recognizer may also be assessed in terms of how close its output is to the correct meaning of the recognized speech. Many authors (e.g. [8] (p. 87) and [78]) have already noticed that the measures like the *WER* are not always a good estimate of understandability of the recognizer's output for the humans.

Speech recognition should always be evaluated in view of the task it is a part of. For example, the real-time factor will be more important for the task of supplying closed captions to live news broadcasting than for the task of building the information retrieval system in the database of audio files with archived broadcast news.

Chapter 7
Experiments and Results

7.1. Text Corpus

7.1.1. Rewriting the Numbers

There are 15% of sentences containing numbers in our corpus. 94% of the corpus measured in bytes is successfully transcribed by our tools. The sentences that failed the transcription are put into separate files, so that new transcription rules can be learned.

7.1.2. Counting the Words

The first thing that is counted in the corpus is its vocabulary. 32 bytes are enough for the computer representation of a single Czech word. Czech words that have more than 31 letters are usually artificially invented and not used in normal utterances. The variable for registering the count of a word in the corpus can be of an integer type that occupies 4 bytes. A single item of the resulting vocabulary of the corpus then occupies 36 bytes.

The words in the corpus are computed according to the following algorithm:

1. A token is identified in the corpus.
2. A list of already found wordforms is searched for this token.
3. If the token is found in this list, its count is incremented by one, otherwise the token is added to this list and its count is initiated to the value of one.

Since this algorithm involves searching in the list of already found wordforms for each token in the corpus, the whole vocabulary of the corpus should be in the operating memory. According to Chapter 6.1.3 we should expect that the vocabulary could have as much as 10 million items. The final data structure of the vocabulary should occupy 360 million bytes, which is equal to 344 MB. This memory space is available in the contemporary personal computers.

The vocabulary of the corpus is the source from which the vocabulary for the recognizer is compiled. Once the vocabulary for the recognizer is available, the n-gram language model can be computed from the corpus. The raw material for the n-gram LM is the list of all short word-successions and their frequencies found in the training corpus. For example, in the case of the bigram LM it is the list of all word-pairs, and in the case of the trigram LM it is the list of all successions of three words. Only such successions are counted in that have all their words present in the recognizer's vocabulary. The largest lexicon for our recognizer compiled so far has 312 thousand words. The number of distinct word-pairs for the bigram LM that have been found in our 2.6-GB training corpus introduced in Chapter 6.1.3 is 60,228,569 [50]. Only those word-pairs that are composed of the words present in our 312-thousand-word vocabulary are counted in that number.

A simple calculation (60,228,569*(2*32+4)) tells us that for the list of 60 million word-pairs and their frequencies 3.8 GB must be allocated in the computer memory. The operating memory of contemporary PCs can be as large as 4 GB, but usual PCs have much smaller operating memory not exceeding 1 GB. It means that counting of word-pairs must be performed in some other way than counting of single words. We propose the following algorithm:

1. A word-pair composed of the words present in the recognizer's vocabulary is identified in the corpus.
2. A sorted list of already found word-pairs is searched for this word-pair using binary search.
3. If the word-pair is found in this list, its count is incremented by one, otherwise an unsorted list of recently found word-pairs is searched for this word-pair.
4. If the word-pair is found in this list, its count is incremented by one, otherwise the word-pair is added to this list and its count is initiated to the value of one.
5. If the unsorted list of recently found word-pairs reaches the number of items equal to *a* then it is sorted together with the already sorted list.
6. If the sorted list reaches the number of items equal to *b* then it is saved into a binary file and deleted.
7. After the whole corpus is processed, all found word-pairs are saved in several sorted binary files.
8. All these files are gradually merged into a single file. Since they are sorted, this operation is neither memory nor CPU intensive. In each step two files are merged. The memory is occupied only by two word-pairs in this process. Each file is read only once.

Figure 7 shows the speed of counting the frequencies of word-pairs (bigrams) and sequences of three words (trigrams) in a 103-MB corpus that contains 16,730,108 tokens. A lexicon of 198,878 words was used to determine how the word-successions should be counted. Every word that was not present in this lexicon was replaced with a special word "OOV". All the resulting word-successions were then counted into the statistics. The time was measured on a 863-MHz PC. It can be seen from this figure that parameter *b* mentioned in point 6 of the above-mentioned algorithm was set to 1 million. The parameter *a* mentioned in point 5 was set to 10 thousand. The required operating memory for the main data structure was then 1,000,000*(2*32+4) B = 65 MB in the case of bigrams and 1,000,000*(3*32+4) B = 96 MB in the case of trigrams.

The upper graph in Figure 7 shows the speed of counting the word-successions. Whenever 1 million new word-successions are found, they are saved into a file, and their list is emptied and filled with newly found word-successions. It can be seen that each million of bigram types is found approximately in 971,110 milliseconds (16 minutes) in the case of bigrams and in 1,071,035 milliseconds (18 minutes) in the case of trigrams. The last bigram token is processed in the 5,864,333th millisecond, and the merging of 7 files lasts 579,042 milliseconds (10 minutes). A file of 4,068,380 word-pairs is the result of this merge. The last trigram token is processed in the 11,738,178th millisecond, and the merging of 11 files lasts

2,050,348 milliseconds (34 minutes). A file of 9,090,817 three-word sequences is the result of this merge.

The lower graph in Figure 7 shows the speed of the same process in terms of the tokens in the corpus. It can be seen from this graph that the process of counting the word-successions is linear regardless of the size of the corpus.

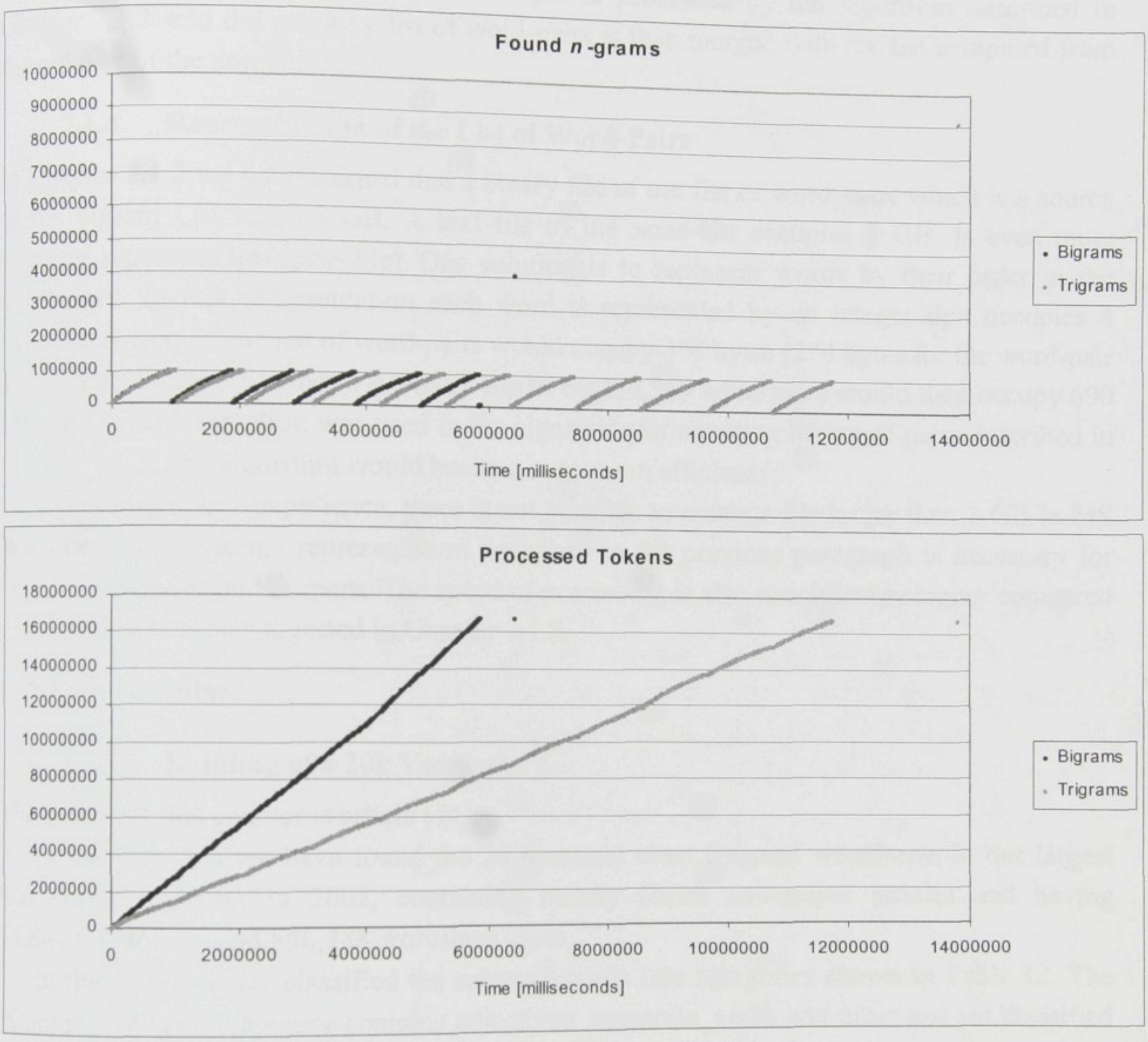


Figure 7. The efficiency of counting words in the corpus

7.1.3. Updating of the List of Word-Pairs

The computation of the list of word-pairs from a several-GB corpus is a considerably time-consuming process. When the corpus is changed by for example applying new rules for its cleaning, the list of word-pairs must be computed from scratch. Very often two different situations appear:

1. The vocabulary of the recognizer is changed.
2. A new part is added to the old corpus.

In Situation 1 the word-pairs that contain some word that is no more in the new vocabulary should be removed from the original list of word-pairs. Then the whole corpus should be processed to get a list of such word-pairs that have at least one word that was added to the new vocabulary and the other word present in the new vocabulary. The resulting two lists will be then merged.

In Situation 2 the new part of the corpus is processed by the algorithm described in Chapter 7.1.2, and the resulting list of word-pairs is then merged with the list computed from the old part of the corpus.

7.1.4. Representation of the List of Word-Pairs

In Chapter 7.1.2 we have learned that a binary file of our list of word-pairs which is a source of the bigram LM has 3.8 GB. A text file of the same list occupies 1 GB. Is even more compact representation possible? One solution is to represent words by their order in the vocabulary. In this representation each word is represented by an integer that occupies 4 bytes. Each item in the list of word-pairs would occupy 3*4 bytes (2*4 bytes for the word-pair and 4 bytes for its frequency). A binary file of 60,228,569 word-pairs would then occupy 690 MB. If this representation was used in the algorithm for counting the word-pairs described in Chapter 7.1.2, this algorithm would become even more efficient.

According to our experience, there is not possible to create a file larger than 2 GB in MS Windows 2000. So, the representation described in the previous paragraph is necessary for large vocabularies and corpora. The speed of processing is also considerably higher compared to the representation suggested in Chapter 7.1.2.

7.2. Vocabulary

7.2.1. Building of a 20k Vocabulary

The source of this chapter is article [54].

In the first step we have found the 20 thousand most frequent wordforms in our largest text corpus existing in 2002, containing mostly Czech newspaper articles and having 55,841,099 tokens and 856,288 wordform types.

In the next step we classified the selected words into categories shown in Table 12. The "Common Words" category contains adjectives, numerals, verbs and other not yet classified words.

Table 12. Frequencies of word categories of the most frequent wordforms of a large corpus [54]

Category	**Number of Wordforms**	**%**
Common Words	18,449	79.96
Pronouns	311	1.35
People's Names	389	1.69
Surnames	777	3.37
Place Names	1,307	5.66
Sport Expressions	289	1.25
Institutions' Names and Trademarks	145	0.63
Political Parties and Other Groups' Names	32	0.14
Currencies' Names	29	0.13
Ambiguous Words	141	0.61
Words with Irregular Pronunciation	541	2.34
Garbage Words	662	2.87
Total Number of Wordforms	**23,072**	**100.00**

To create our vocabulary from these classified words, we eliminated garbage words, some less important place names and foreign names, and surnames. After that we have made a process that could be called equalization. For example, in the case of numerals equalization consists in determining what numbers should represent numerals in our vocabulary, what are the most common inflections of numerals, that have made it into the 20 thousand most frequent words of our corpus, and supplying the missing most common inflections into the whole scale of numerals. The same process should be done with other word categories.

We have found that people's names, surnames, and place names occur mostly in nominative and genitive cases. The most frequent people's names and surnames occur also in dative and instrumental cases, and the most frequent place names occur also in local case. We supplied these most frequent grammatical cases of all Czech people's names and important Czech place names and all missing pronouns into our vocabulary.

The resulting vocabulary should be tested on some independent text corpus in order to get a percentage of words in this corpus that are covered by the vocabulary. Words that were not found in the vocabulary should be examined and supplied into the vocabulary provided they are not too specific. This could be done in several iterations.

We enriched our vocabulary by missing important common words according to the following procedure. We have collected two corpora, each having more than 3 million tokens that are not a part of our 55,841,099-word corpus. One of them contained the Internet version of the newspaper "Lidové noviny" (52% of corpus) and the Internet newspaper "Neviditelný pes" (17%), 4 diploma theses (2%), and 27 novels (29%). The other corpus consisted of science fiction novels. We have found 85,011 wordforms that were present in both corpora. This set of words is to a certain extent stripped of the words that are specific to each of source corpora respectively. 66,674 wordforms of this set were not present in our previous version of vocabulary. We sorted these words according to their joint frequencies in both corpora to get the most frequent of them. We have examined these most frequent words and supplied 284 of them that were subjectively felt as important, into our vocabulary. This was the final step in its creation. The final version of our vocabulary had 23,514 wordforms.

We have also found 3,820 wordforms in our vocabulary that were not present in the 85,011-word intersection of the two corpora. Exploration of these words shows that they are mostly specific for Czech news like Czech place names, sport expressions, institutions' names and trademarks, political parties names, and some common terms frequently used in news. So, the removal of these words could help to optimize our vocabulary for non-Czech-journalist use.

The final 23,514-word version of our vocabulary was tested on another corpus. This corpus, collected from electronic version of the popular Czech magazine about politics and culture "Reflex", has 732,569 words and 100,866 wordforms. 80% of tokens and 20% of wordform types of it was covered (present) in our final version of vocabulary.

7.2.2. Building of a 800k Vocabulary and the Analysis of Text Coverage

The source of this chapter are articles [56] and [12].

The training corpus for obtaining our 800k vocabulary has been compiled from our previously collected corpus of newspaper and novel texts introduced at the beginning of Chapter 7.2.1 and complemented by approximately the same number of sentences taken from the Czech National Corpus [57]. After eliminating virtually all the typing errors using the spellchecker [55] the resulting corpus had 135,661,782 tokens. The vocabulary of this corpus converted to the lower-case letters had 644,635 wordform types. These words were merged with the synthetically generated set (see Chapter 6.2.4), which resulted in an inventory of 788,274 different wordforms. This set is our 800k vocabulary.

The test corpus for the computation of coverage of our 800k vocabulary has already been mentioned in Chapter 7.2.1. This corpus consisted of the Internet version of "Lidové noviny" (52% of corpus), the Internet newspaper "Neviditelný pes" (17%), 4 diploma theses (2%), and 27 novels (29%). We have removed from this corpus short nonsensical words and abbreviations that usually appear only in written language in the amounts of 3,441 wordforms and 51,781 tokens. The resulting test corpus had 3,034,108 tokens and 169,381 wordform types.

141,865 wordforms (2,949,762 tokens) of the test corpus were found in our 800k vocabulary. The rest – 27,516 wordforms (84,346 tokens) – were not covered by our largest vocabulary. From these figures it can be computed that more than 97% of tokens and more than 83% of wordforms of the test corpus are covered (present) in our 800k vocabulary.

Our 800k vocabulary and the independent corpus were used to compute a coverage curve for the Czech language. This curve should answer two questions: How many most frequent wordforms are enough to attain a reasonable coverage, and, provided we have a vocabulary of a certain number of words, what percentage of coverage of an independent speech should we expect? The curve is shown in Figure 8.

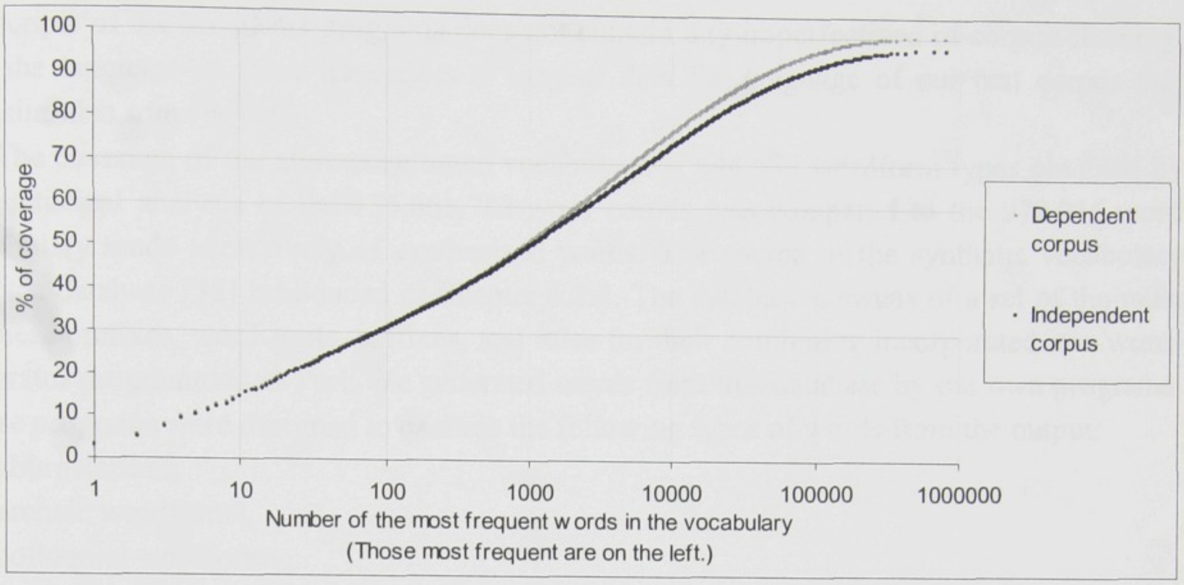


Figure 8. Coverage of the dependent and the independent corpus [56]

The dependent corpus is the training corpus of the 800k vocabulary. Each measurement in Figure 8 has been done for the group of words with the same frequency in the vocabulary. The words in the vocabulary have been sorted by frequencies from the most to the least frequent ones. The horizontal axis in Figure 8 is logarithmic, because in this way the coverage of smaller vocabularies is especially transparent.

Figure 9 shows the detail of the same coverage curve for the vocabularies with the number of words between 100,000 and 800,000. The horizontal axis in Figure 9 is non-logarithmic

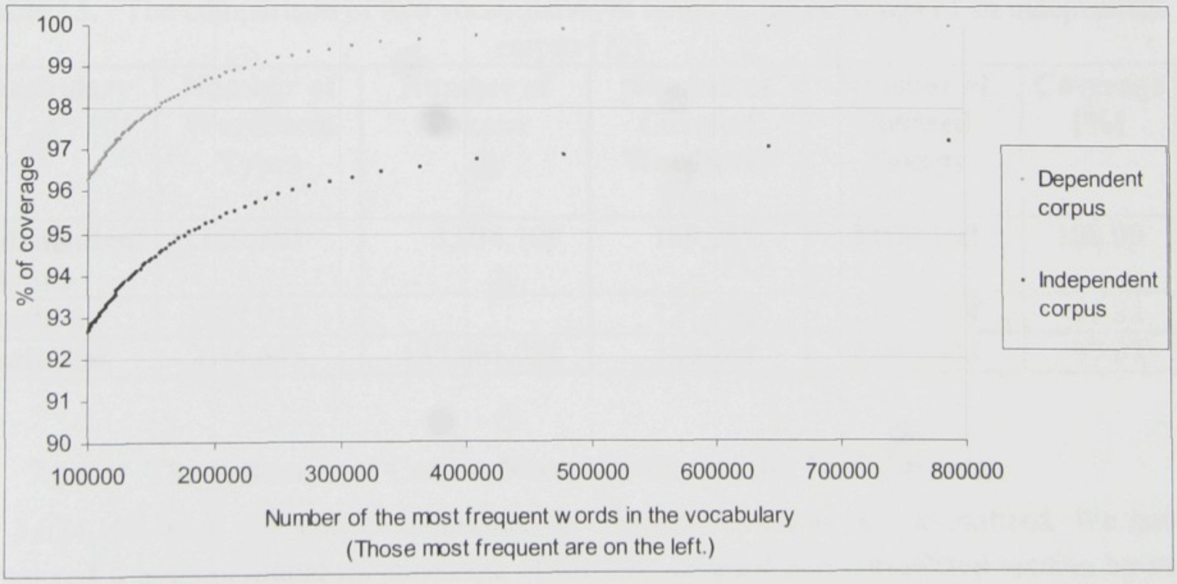


Figure 9. Coverage of the dependent and the independent corpus [56]

While Figure 8 and Figure 9 show the coverage of the corpus of written language, article [50] compares the coverage of written language and spoken language. The spoken language in this article is represented by approximately 3 MB of our transcripts of various types of broadcast programs. A 300k lexicon can cover 98% and 800k lexicon covers as much as 99% of spoken language. There are two reasons for these optimistic results: The corpus of the transcripts of the broadcast programs does not contain any imperfections of corpus cleaning, and the language of these transcripts is simpler than the language of our text corpus that contains also some novels.

The coverage of the abovementioned vocabulary of 644,635 wordform types obtained by the statistical analysis of the 135,661,782-word corpus was compared to the 972,915-word vocabulary made exclusively of synthesized words. The source of the synthetic vocabulary was the database [52] introduced in Chapter 6.2.4. The database consists of a set of the most frequent prefixes, word-roots, suffixes, and rules for their combining incorporated in a word-generator programmed in Perl. We generated words from this database by our own programs. These programs were designed to exclude the following types of words from the output:

1. abbreviations,
2. archaic wordforms,
3. colloquial wordforms,
4. specialized terms, e.g. chemicals,
5. plural number of personal names and place names.

The source database had 74,867 roots and 5,729 endings. 3,834,170 wordforms were generated from it but because of a great number of duplicates only 972,915 wordforms were unique. For example, some adjectives had 27 identical wordforms, each having a distinct morphological feature.

The text collection used for the measuring of coverage was the abovementioned 3,034,108-word corpus. The results of the comparison of coverage of the both vocabularies are shown in Table 13.

Table 13. The comparison of two vocabularies in terms of the coverage of an independent corpus [12]

Vocabulary	Number of Wordform Types	Number of Tokens	Number of Covered Wordform Types	Number of Covered Tokens	Coverage [%]
Independent Corpus	169,381	3,034,108	169,381	3,034,108	100.00
Synthetic	972,915	.	121,592	2,907,154	95.82
Statistical	644,635	135,661,782	140,231	2,945,335	97.07

7.2.3. The Influence of Corpus Normalization on the Text Coverage

Our 3,034,108-word test corpus mentioned in Chapter 7.2.2 was not normalized. We have normalized it and computed the coverage of the both original and normalized version by our 312k vocabulary. The process of normalization is explained in point 8 in Chapter 6.1.2. The results in Table 14 show that the coverage of the normalized corpus is higher by 0.3% and that the vocabulary of 169k words can be reduced by approximately 1,000 wordforms thanks to the normalization.

Table 14. The comparison of the coverage of the test corpus with its normalized version by the 312k lexicon

	Number of Wordform Types	Number of Tokens	Number of Covered Wordform Types	Number of Covered Tokens	Coverage [%]
Original Corpus	169,381	3,034,108	122,643	2,928,731	96.53
Normalized Corpus	168,352	3,034,108	123,235	2,938,345	96.84

7.2.4. The Influence of the Vocabulary Size on Broadcast News Transcription

The speech data used for the comparison of accuracy of BNT attained with our 312k lexicon with its downscaled versions were three complete TV shows from three different Czech stations first mentioned in Chapter 3.1.

The Rank value in Table 15 is the threshold value of frequency $\hat{C}(w)$ from formula (5) in Chapter 6.2.9 that all words in the particular lexicon must exceed.

Table 15. Recognition rates achieved with lexicons of different sizes [15]

Name	Size [words]	Rank	OOV [%]	Accuracy [%]
Lex64k	64,620	300	5.17	70.96
Lex102k	102,228	140	3.31	73.75
Lex149k	148,928	70	1.94	75.62
Lex195k	194,932	40	1.34	76.64
Lex257k	257,086	20	0.97	77.27
Lex312k	312,289	10	0.64	78.13

7.3. Phonetic Transcription

7.3.1. The Importance of Exceptions in Phonetic Transcriptions

In 2004 we increased our 140k vocabulary into the 200k vocabulary. The new 60k words were transcribed by our rule-based system for phonetic transcription that could transcribe correctly all regular Czech words and few exceptions. The WER of CSR with the resulting 200k vocabulary was higher than the WER using the original 140k vocabulary. Only after many manual corrections of phonetic transcriptions of words that contain exceptions from the regular Czech pronunciation and some foreign words in the 200k vocabulary the WER dropped under the original level in the 140k vocabulary. This case is mentioned in [51]. Article [68] informs that our then existing rule-based system was transcribing 92.4% words in our dictionary correctly.

It was a motivation to improve our rule-based system for phonetic transcription, so that it could cope with more Czech exceptions from the regular pronunciation. We needed to compile a better list of its rules. The method that was used is described in Chapter 6.3.6.

7.3.2. The Influence of Additional Phonetic Transcriptions

The first experiment that assessed the importance of additional phonetic transcriptions in our 200k vocabulary was reported in paper [13]. In this experiment 2,358 additional phonetic transcriptions improved the accuracy of recognition from 70.85% to 71.53%.

Article [50] reports the similar experiment for the 310k lexicon. In this experiment 19,890 additional phonetic transcriptions improved the accuracy of recognition from 75.5% to 76.9%.

The latest experiment of that kind for the 312k lexicon is presented in paper [15]. In this experiment 18,933 additional phonetic transcriptions improved the accuracy of recognition from 77.06% to 78.13%.

These three experiments were obtained on the same speech data – three complete TV shows from three different Czech stations first mentioned in Chapter 3.1.

7.3.3. The Influence of Glottal Stop

The experiment that assessed the importance of the phoneme called "glottal stop" (see Table 3 in Chapter 6.3.3) is published in paper [15]. The removal of glottal stop from all phonetic transcriptions in our 312k vocabulary decreased the accuracy of recognition from 78.13% to 77.92%.

7.4. Language Model

7.4.1. The Influence of Various Language Models on Continuous Speech Recognition of a Development Test Set Using a Closed Vocabulary

In 2001 members of the SpeechLab carried out their first experiments with CSR supported by the bigram LMs. The setup of these experiments is given in Chapter 3.1. The Zerogram, Unigram, and Add-One LMs in Table 16 are computed from our 55-milion-word independent training corpus. The Maximum Likelihood Estimate and Witten-Bell LMs are computed both from our training corpus and from the development test set of 800 sentences used for the recognition tests.

Table 16. Recognition rates achieved for different LM types based either on an independent text corpus or on the test set of 800 sentences. The perplexity is conditional cross perplexity of the LM against the development test set. The vocabulary size was 3,622 words. [10]

Language Model Type	Perplexity	Accuracy (%)	Sentence Recognition Rate (%)
Corpus based – Zerogram	3,622	54.2	3.5
Corpus based – Unigram	3,149	54.4	2.9
Corpus based – MLE-2	54	56.1	3.8
Corpus based – Add1-2	572	58.7	7.3
Corpus based – WB-2	417	63.9	9.1
Test based – MLE-1	5	82.3	37.5
Test based – WB-1	10	76.7	26.6

In 2002 the experiments with the same development test set were repeated. We have improved the acoustic models and found a better set of *LM Factor*, *Word Insertion Penalty*, *Prune Threshold*, and *Number of Word-End Hypotheses* parameters discussed in Chapter 3.2. The results are shown in Table 17.

Table 17. Recognition rates achieved for different LM types based either on an independent text corpus or on the test set of 800 sentences. The vocabulary size was 3,622 words. For each LM type the optimal combination of the weight of the language model – the *LM Factor* and the *Word Insertion Penalty* was found to achieve the highest Accuracy. The other parameters were constant: *Prune Threshold* = 100, and *Number of Word-End Hypotheses* = 7. [11]

Language Model Type	Accuracy (%)	LM Factor	Word Insertion Penalty
Corpus based – Zerogram	50.85	0	– 35
Corpus based – MLE-2	48.42	5	– 11
Corpus based – Add1-2	62.98	5	– 7
Corpus based – WB-2	66.63	6	– 6
Test based – MLE-1	94.50	2	– 2

The results in Table 17 underscore the importance of LM smoothing. The Add One and Witten-Bell LMs outperform the Maximum Likelihood Estimate LM (MLE-2 that was a source of their computation) more distinctly than in Table 16. The accuracy of the MLE-2 LM was even lower than the accuracy attained without any LM (the Zerogram). The accuracy attained with the MLE-1 LM computed from the development test set has shown the upper limits of our continuous speech recognition system existing in 2002.

The same year the same experiment was carried out with the same test set extended by another 800 sentences. The results are shown in Table 18.

Table 18. Recognition rates achieved for different LM types based either on an independent text corpus or on the test set of 1,600 sentences. The vocabulary size was 7,033 words. For each LM type the optimal combination of the weight of the language model – the *LM Factor* and the *Word Insertion Penalty* was found to achieve the highest Accuracy. The other parameters were constant: *Prune Threshold* = 130, and *Number of Word-End Hypotheses* = 10. [17]

Language Model Type	Accuracy (%)	LM Factor	Word Insertion Penalty
Corpus based – Zerogram	48.63	0	– 42
Corpus based – MLE-2	47.52	5	– 12
Corpus based – Add1-2	61.58	5	– 3
Corpus based – WB-2	65.48	6	– 5
Test based – MLE-1	95.82	2	– 9

7.4.2. The Influence of Collocations on Broadcast News Transcription

The speech data used for the experiments with collocations were three complete TV shows from three different Czech stations introduced in Chapter 3.1.

Article [13] reports that adding 314 collocations shorter than three syllables into the 200k lexicon increased the accuracy of recognition from 71.11% to 71.53%.

Article [50] reports that adding 1,708 collocations into the 310k lexicon increased the accuracy of recognition from 75.5% to 78.2%.

7.4.3. The Influence of Word-Decomposition on Broadcast News Transcription

The lexicon used for the experiment assessing the effect of word-decomposition had 312,211 word-forms. Its decomposed version had 290,480 word-forms. In the process of the creation of the decomposed lexicon a list of 128,995 word-forms that could be decomposed was compiled. This list contains the original word-forms together with the information how these word-forms should be decomposed. This list was used to decompose the list of word-pairs found in the training corpus to obtain the decomposed language model for the 290,480-word vocabulary.

The same list was used in the phase of post-processing of the output of the recognizer. The recognizer that uses the decomposed lexicon outputs decomposed words, e.g. instead of the word *pětapadesátiletý* (*55-years old*) it writes *pět a padesáti letý*. All sequences of 4, 3, and 2 words in the output of the recognizer were concatenated when their concatenation was found in the list of 128,995 words to be decomposed. After this processing the standard procedure using the minimum edit distance algorithm was run to obtain the accuracy of recognition. The results are in Table 19.

Table 19. The results of BNT when using the vocabulary of whole word-forms compared with the results obtained with the vocabulary of decomposed words. The information about test speech databases used in this task is in Chapter 7.5.

Test Speech Database	**Accuracy [%]**		**Correctness [%]**		**OOV Rate [%]**	
	Original LM	**Decomposed LM**	**Original LM**	**Decomposed LM**	**Original LM**	**Decomposed LM**
TV News	78.539	77.900	80.871	80.185	0.68655	0.63920
Radio News	81.676	80.957	84.076	83.465	1.78121	1.72770
Total	80.445	79.758	82.819	82.178	1.35167	1.30057

The results in Table 19 show that word-decomposition has not improved the accuracy of recognition even if it has improved the OOV rate. A closer look at the results shows that the degradation in the accuracy was not definite. In 124 utterances the accuracy with the decomposed LM was lower, but in 45 utterances the word-decomposition increased the accuracy. Examination of some sentences suggests that the reason why word-decomposition was not very successful are the same as the reasons (see Chapter 6.2.10) why we have introduced the collocations concatenated into single words into our lexicon. Chapter 7.4.2 shows that collocations concatenated into single words have definitely improved the accuracy of BNT.

Additional information about language models used for the experiments described in this chapter is in Table 20. Entropy in this table was computed according to formula (34) in Chapter 6.4.11. The LMs were smoothed using the improved Witten-Bell discounting proposed in Chapter 6.4.6 in formulas (14), (15), (20), (25), and (26).

Table 20. The description of language models used in the study of the effect of word-decomposition

Language Model	Vocabulary Size	Entropy	Perplexity	Number of Tokens	Number of Word-Pair Types
Original	312,211	11.061663	2,137	368,037,760	62,614,572
Decomposed	290,480	10.841029	1,834	383,613,230	59,681,588

We have also used the text part of our two well-annotated speech databases to test the appropriateness of three methods of LM smoothing discussed in Chapter 6.4.6. The result can be seen in Table 21, and it suggests that the Witten-Bell discounting improved by formulae (25) and (26) is the best, because the conditional cross perplexity of LM smoothed that way against the both test corpora is the lowest.

Table 21. Conditional cross perplexity of three variants of Witten-Bell discounting

Test Speech Database	Original LM			Decomposed LM		
	Classical Witten-Bell	Witten-Bell with Add-One Smoothing	Improved Witten-Bell	Classical Witten-Bell	Witten-Bell with Add-One Smoothing	Improved Witten-Bell
TV News	859.040	859.056	858.511	775.787	775.818	775.372
Radio News	704.251	704.098	703.770	643.252	643.131	642.837

7.5. Test Speech Databases

The results of this thesis are computed on three test databases.

The first speech database consisted of 800 sentences first mentioned in Chapter 3.1. Later this database was enlarged by another set of 800 sentences and results attained on this database together with its description were published in [16] and [17]. The database of 1,600 sentences has 16,027 word tokens.

The second speech database referred to in this thesis consists of 3 complete TV shows. It is also first mentioned in Chapter 3.1 and its description can be found in [13]. This database has 8,451 tokens.

The third speech database was created for the purpose of this thesis and for possible future research. Its source is radio news broadcasting of public Czech station Radiožurnál and British public station BBC whose broadcasting in Czech was ended in 2005. The description of this database that has 13,081 tokens is in Table 22 and Table 23. The database contains utterances from 61 speakers. 14 utterances contain some slip of the tongue.

Table 22. Number of utterances = sentences = sound files in the speech database of radio news

Radio Station	Gender	Speaking Style	Sound Condition			Total	Total
			Clear	Low Fidelity (e.g. phone)	Background Noise (speech, music)		
Radiožurnál	male	professional	76	8	33	117	436
Radiožurnál	male	guest	49	27	4	80	
Radiožurnál	female	professional	196	6	30	232	
Radiožurnál	female	guest	1	5	1	7	
BBC	male	professional	124	0	27	151	411
BBC	male	guest	22	50	8	80	
BBC	female	professional	160	0	20	180	
BBC	female	guest	0	0	0	0	
Total	male		271	85	72	428	
Total	female		357	11	51	419	
Total	professional		556	14	110	680	
Total	guest		72	82	13	167	
Total			628	96	123	847	847

Table 23. Number of seconds in the speech database of radio news

Radio Station	Gender	Speaking Style	Sound Condition			Total	Total
			Clear	Low Fidelity (e.g. phone)	Background Noise (speech, music)		
Radiožurnál	male	professional	370	41	146	557	2,558
Radiožurnál	male	guest	426	248	25	698	
Radiožurnál	female	professional	1,063	31	141	1,235	
Radiožurnál	female	guest	5	50	13	68	
BBC	male	professional	807	0	139	946	2,852
BBC	male	guest	209	516	108	833	
BBC	female	professional	973	0	101	1,073	
BBC	female	guest	0	0	0	0	
Total	male		1,812	804	417	3,033	
Total	female		2,041	81	255	2,377	
Total	professional		3,212	71	527	3,811	
Total	guest		640	814	145	1,599	
Total			3,853	885	672	5,410	5,410

7.6. Tuning of the Recognizer's Parameters

The first series of our tuning experiments with our CSR recognizer were carried out in 2002. The devtest set consisted of 800 sentences. Figure 10 shows the influence of various values of the *Word Insertion Penalty* parameter ceteris paribus (the other parameters being fixed) on the accuracy of recognition. The detail on the left upper side of Figure 10 shows that the *Word Insertion Penalty* parameter has some optimal value that is in this case – 35.

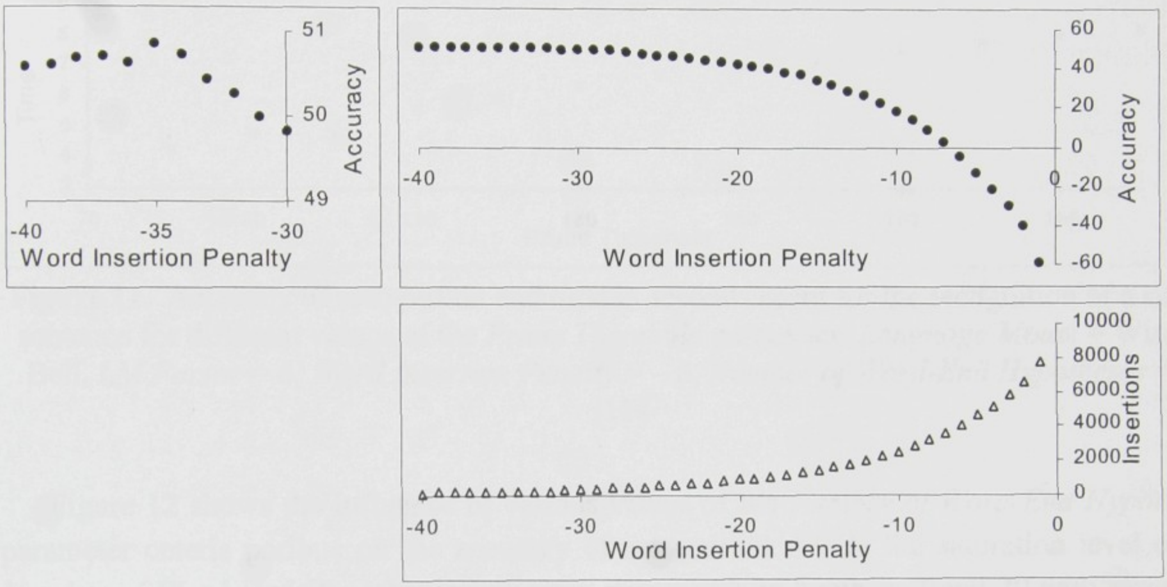


Figure 10. Accuracy of recognition and the number of false word insertions for different values of the *Word Insertion Penalty* parameter. *Language Model* = Zerogram, *Prune Threshold* = 100, *Number of Word-End Hypotheses* = 7. [11]

Figure 11 shows the influence of various values of the *Prune Threshold* parameter ceteris paribus on the accuracy of recognition. While the saturation level of the *Prune Threshold* parameter is somewhere between 100 and 150 with respect to the accuracy, the time consumption rises linearly in proportion with the varying parameter.

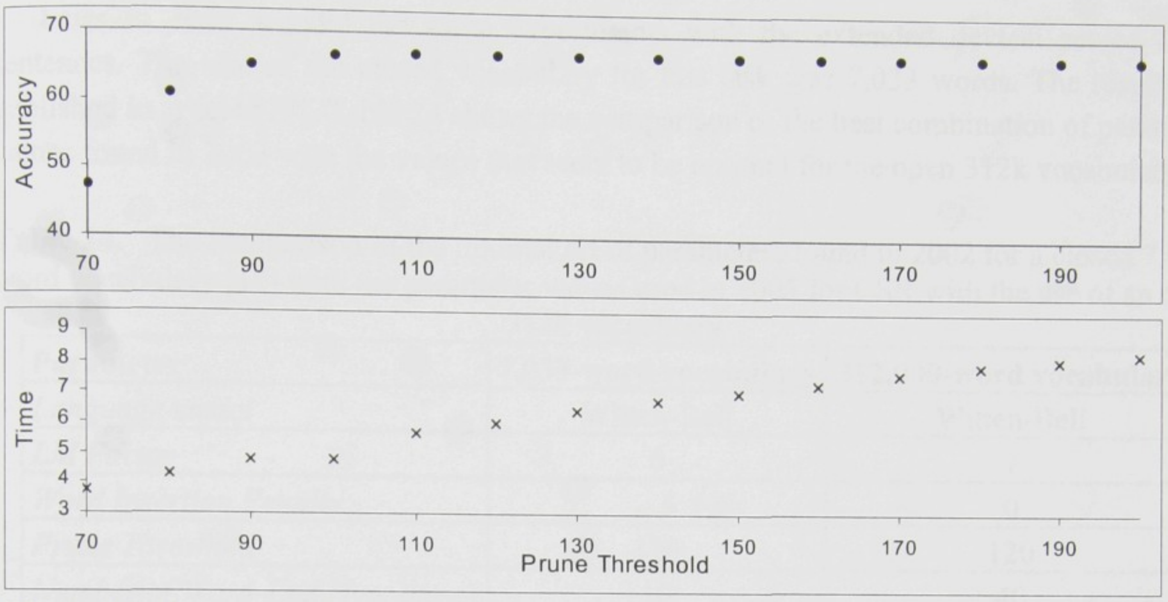


Figure 11. Accuracy of recognition and time in seconds spent for the recognition of a single sentence for different values of the *Prune Threshold* parameter. *Language Model* = Witten-Bell, *LM Factor* = 6, *Word Insertion Penalty* = – 6, *Number of Word-End Hypotheses* = 7. [11]

Figure 12 shows the influence of various values of the *Number of Word-End Hypotheses* parameter ceteris paribus on the accuracy of recognition. While the saturation level of the *Number of Word-End Hypotheses* parameter is somewhere between 7 and 20 with respect to the accuracy, the time consumption rises also linearly in proportion with the varying parameter as it did in Figure 11.

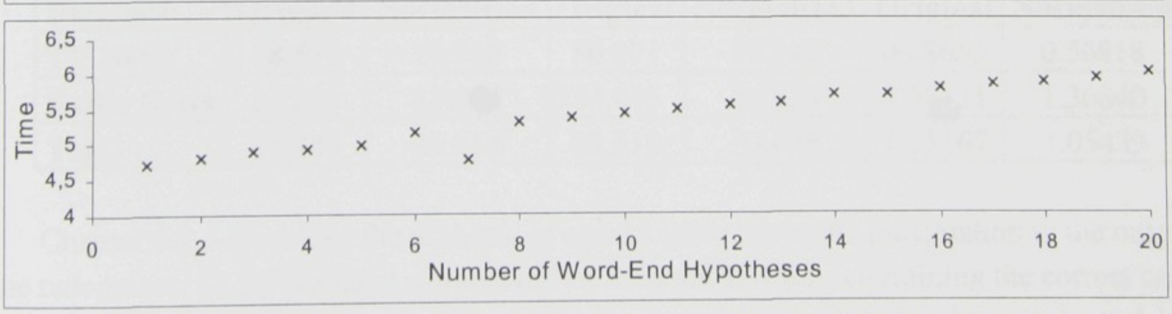


Figure 12. Accuracy of recognition and time in seconds spent for the recognition of a single sentence for different values of the *Number of Word-End Hypotheses* parameter. *Language Model* = Witten-Bell, *LM Factor* = 6, *Word Insertion Penalty* = – 6, *Prune Threshold* = 100. [11]

Later in 2002 we did the same experiments with the extended devtest set of 1,600 sentences. The size of the closed vocabulary for this task was 7,033 words. The results are published in article [17]. Table 24 shows the comparison of the best combination of parameter values found in 2002 with the values that seem to be optimal for the open 312k vocabulary.

Table 24. The comparison of the optimal set of parameters found in 2002 for a closed 7,033-word vocabulary [17] with the parameter values used in 2005 for CSR with the use of an open 312k vocabulary

Parameter	7,033-word vocabulary	312,000-word vocabulary
Language model	Witten-Bell	Witten-Bell
LM Factor	6	7
Word Insertion Penalty	– 5	0
Prune Threshold	130	120
Number of Word-End Hypotheses	10	40

The differences in the optimal parameter values in Table 24 can be explained by the substantial increase in the vocabulary size. In this case the average probability in the LM decreases and so the weight of the LM (the *LM Factor*) should be enlarged, and a larger number of words must be taken into consideration in the course of the recognition, hence the substantial increase in the *Number of Word-End Hypotheses* and maybe also the change in the *Word Insertion Penalty* parameter.

7.7. Evaluation of Recognition

This chapter presents several results that could be derived from the experiments described in Chapter 7.4.3.

The first result shows the improvement in the accuracy of recognition when the reference transcriptions are normalized according to point 8 in Chapter 6.1.2.

Table 25. The importance of normalization of reference transcriptions into standard orthography. The results superscribed as "original" are taken from Table 19.

Test Speech	Accuracy [%]		Correctness [%]		OOV Rate [%]	
Database	Original	Normalized	Original	Normalized	Original	Normalized
TV News	78.539	78.610	80.871	80.942	0.68655	0.56818
Radio News	81.676	81.974	84.076	84.374	1.78121	1.36840
Total	80.445	80.654	82.819	83.028	1.35167	1.05439

Chapter 6.1.5 discusses the problem of correct letter cases and punctuation in the output of the recognizer. So far we have adopted only a unigram LM for determining the correct case of letters in the output of the recognizer. The accuracy of recognition when upper and lower letter cases are distinguished is 80.249% in comparison to 81.676% (see the original accuracy of the radio news test speech database in Table 25). A closer look at the mistakes in placement of correct letter cases suggests that introduction of a bigram LM for this task would solve at least 90% of mistakes existing now. This LM should also contain punctuation marks, because many mistakes were caused by the fact that a single utterance contained more than

one sentence. The first letters of each utterance were always converted to the upper case before the computation of accuracy.

Table 26 shows the accuracy of recognition divided into categories of Table 22 and Table 23 in Chapter 7.5.

Table 26. An analytical view of the accuracy in percent of the basic recognition experiment on the test speech database of radio news (all words converted to lower case and an undecomposed LM).

Radio Station	**Gender**	**Speaking Style**	**Sound Condition**			**Total**	**Total**
			Clear	**Low Fidelity (e.g. phone)**	**Background Noise (speech, music)**		
Radiožurnál	male	professional	87.851	81.053	77.635	84.730	82.827
Radiožurnál	male	guest	76.739	54.869	78.689	69.109	
Radiožurnál	female	professional	90.709	83.099	83.668	89.768	
Radiožurnál	female	guest	83.333	69.748	8.571	58.721	
BBC	male	professional	83.006	.	75.749	81.850	80.589
BBC	male	guest	73.036	67.130	78.298	70.043	
BBC	female	professional	87.681	.	80.465	87.077	
BBC	female	guest	.	.	.	.	
Total	male		81.703	64.193	77.186	76.684	
Total	female		89.295	74.737	78.130	87.700	
Total	professional		87.618	81.928	79.167	86.349	
Total	guest		75.586	63.672	70.997	69.094	
Total			85.826	65.211	77.529	81.676	81.676

One of the frequent problems in BNT is a fast speaking rate. Speaking rate means how many words or phones are uttered at a fixed time interval. The dependency of speaking rate and the accuracy of recognition on our radio news test speech database is shown in Figure 13. The graph suggests that our recognizer has no problem with speaking rate. If it had had a lower accuracy of faster speaking rate, we would have to make some adjustments on the signal level of recognition as proposed e.g. in [75].

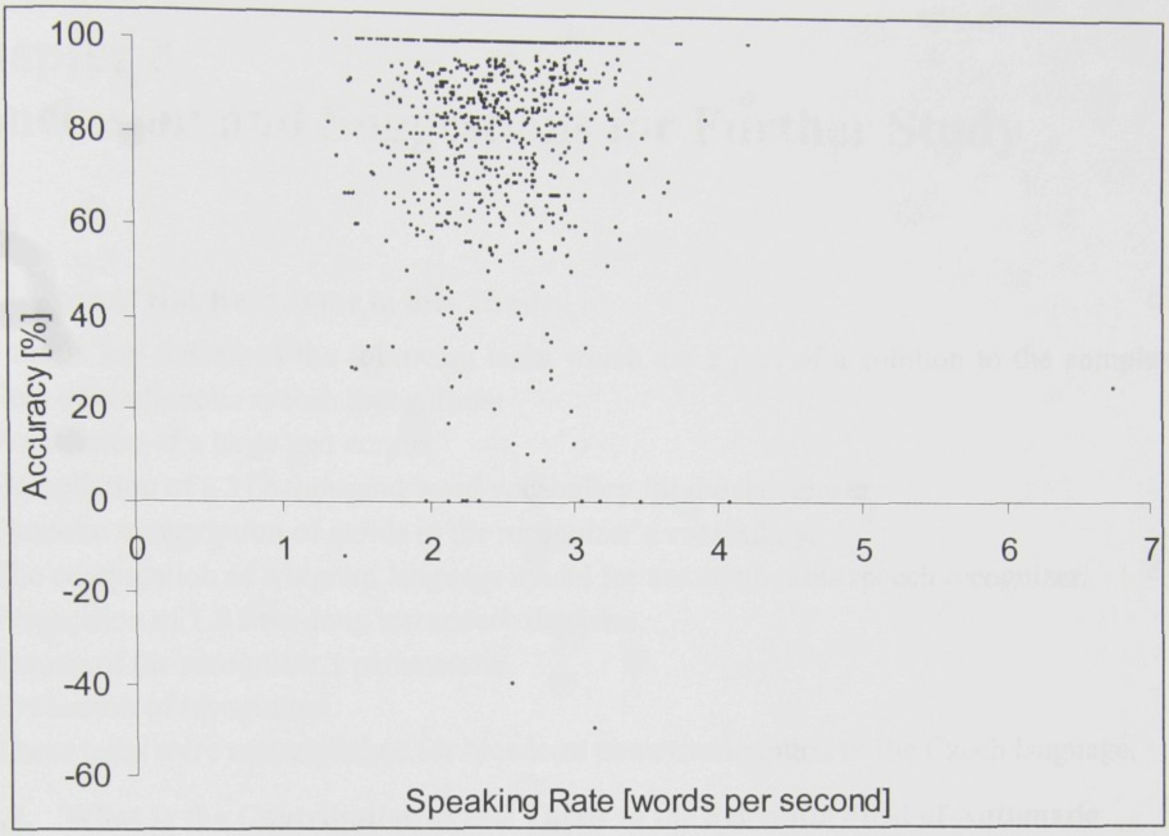


Figure 13. The dependency of the accuracy of recognition on the speaking rate

Chapter 8
Conclusion and Suggestions for Further Study

8.1. What Has Been Done in this Thesis

This thesis has described the following tasks which are a part of a solution to the complex problem of continuous speech recognition:

1. Preparation of a large text corpus,
2. Compilation of a 312-thousand-word vocabulary for the recognizer,
3. Phonetic transcription of words in the recognizer's vocabulary,
4. The computation of a bigram language model for our continuous speech recognizer,
5. Preparation of 1.5 hour-long test speech database,
6. Tuning of the recognizer's parameters,
7. Evaluation of recognition.

These tasks were accomplished for broadcast news transcription of the Czech language.

8.2. What Is the Contribution of this Thesis to the Scientific Field of Automatic Speech Recognition

This thesis has described the tasks specified in Chapter 8.1 in detail that is not usual in articles dealing with automatic speech recognition.

A vocabulary of 800 thousand most frequent Czech wordforms was compiled to get a robust estimation of independent text coverage by lexicons of various sizes for the case of the Czech language.

Three alternative methods of phonetic transcriptions were described, and recognition experiments have confirmed the importance of correct phonetic transcriptions in the recognizer's vocabulary.

The Witten-Bell discounting used for the smoothing of the bigram language model was enhanced and the resulting cross perplexity of the smoothed language model against test corpus was indeed lower than that of the language models smoothed by alternative smoothing methods.

It was discovered that decomposition of words into their parts in the recognizer's vocabulary and language model does not in average increase the accuracy of recognition. However, the inverse approach – joining words that often stand together in text into single words – has significantly improved the accuracy of recognition.

8.3. What Is the Contribution of this Thesis to the Practice

The results of work described in this thesis – text corpus, vocabulary, phonetic transcription, language model, and test speech database – are points of departure for future improvements of SpeechLab's continuous speech recognizer.

The thesis has shown that the approaches described in it can probably lead to some commercially usable applications of broadcast news transcription of the Czech language.

8.4. What Should Be Done in the Future

The increase in the accuracy of recognition is a result of work in many fields. Here we mention only the tasks that concern the linguistic part of the problem.

1. More rules for the cleaning of text corpus should be compiled, namely the rules for rewriting of numerals.
2. Table 15 in Chapter 7.2.4 shows that the addition of one or two hundred thousand new words into our 312-thousand-word vocabulary for the recognizer could still improve the accuracy of recognition.
3. Our system for phonetic transcription should contain more and better organized rules. It should also be able to generate alternative phonetic transcriptions.
4. The language models for our recognizer have always been limited by the capacity of the currently available hardware of personal computers. However, the development of hardware is so fast that we should design new language models implementable on future hardware as early as today. The most promising approach in our opinion is to develop language models that could be automatically focused on the current topic of speech that is being recognized.
5. The output of the recognizer should be as close to the written text as possible. This involves placement of correct letter cases and punctuation. Also some numerals should be written using digits and some of them should be written using words. Special language models must be compiled to solve this problem.

Bibliography

[1] http://www.pcai.com/Paid/Issues/PCAI-Online-Issues/16.6_OL/New_Folder/Sample_16.6/PCAI-16.6-Sample-pg.18-Art1.htm
cited on June 7, 2005.

[2] http://www.dragon-medical-transcription.com/historyspeechrecognitiontimeline.html
cited on June 7, 2005.

[3] Stuart Russell, Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall International, Inc., New Jersey, 1995, ISBN 0-13-360124-2, p. 770.

[4] Josef Chaloupka: Rozpoznávání akustického signálu řeči s podporou vizuální informace. [Disertační práce]. Liberec 2005. – Technická univerzita v Liberci. Fakulta mechatroniky a mezioborových inženýrských studií.

[5] http://www.informedia.cs.cmu.edu/
cited on June 7, 2005.

[6] Daniel Jurafsky, James H. Martin: Speech and Language Processing. Prentice Hall, Inc., New Jersey, 2000, ISBN 0-13-095069-6.

[7] Jan Nouza (editor): Počítačové zpracování řeči, cíle, problémy a aplikace. (Sborník článků). Technická univerzita v Liberci. Fakulta mechatroniky a mezioborových inženýrských studií. Katedra elektroniky a zpracování signálů – Laboratoř počítačového zpracování řeči. Liberec 2001, ISBN 80-7083-551-6.

[8] Pavel Ircing: Large Vocabulary Continuous Speech Recognition of Highly Inflectional Language (Czech). [Dissertation thesis] University of West Bohemia in Pilsen. Faculty of Applied Sciences. Plzeň 2003.

[9] http://itakura.kes.vslib.cz/kes/indexe.html
visited on July 27, 2005.

[10] Jan Nouza, Dana Nejedlová: Experiments with Read Speech Recognition in Czech. In: Speech Processing, 11th Czech-German Workshop on Speech Processing, Prague 2001, pp. 46 – 49, ISBN 80-86269-07-8.

[11] Dana Nejedlová, Jan Nouza: Language Model Support for Continuous Speech Recognition in Czech Language. In: Signal Processing, Pattern Recognition, and Application, Anaheim (USA), Calgary (Canada), Zurich (Switzerland) 2002, ISBN 0-88986-338-5, pp. 541 – 546, ISSN 1482-7921.

[12] Dana Nejedlová, Jan Nouza: Building of a Vocabulary for the Automatic Voice-Dictation System. In: Text, Speech and Dialogue (eds. Václav Matoušek; Pavel Mautner), Springer-Verlag, Heidelberg, 2003, pp. 301 – 308, ISBN 3-540-20024-X, ISSN 0302-9743.

[13] Jan Nouza, Dana Nejedlová, Jindřich Žďánský, Jan Kolorenč: Very Large Vocabulary Speech Recognition System for Automatic Transcription of Czech Broadcast Programs. In: Proceedings of ICSLP (eds. Soon Hyob Kim and Dae Hee Youn), Sunjin Printing Co., 2004, pp. 409 – 412, ISSN 1225-441x.

[14] Jan Nouza, Jindřich Žďánský, Petr David: Fully Automated Approach to Broadcast News Transcription in Czech Language. In: Text, Speech and Dialogue (eds. Petr Sojka, Ivan Kopeček, Karel Pala), Springer-Verlag, Heidelberg, 2004, pp. 401 – 408,

[15] Dana Nejedlová, Jindra Drábková, Jan Kolorenč, Jan Nouza: Lexical, Phonetic, and Grammatical Aspects of Very-Large-Vocabulary Continuous Speech Recognition of Czech Language. In: Electronic Speech Signal Processing, Proceedings of the 16th Conference on Electronic Speech Signal Processing joint with the 15th Czech-German Workshop on Speech Processing, Dresden, Germany, September 2005, TUDpress, pp. 224 – 231, ISBN 3-938863-17-X, ISSN 0940-6832.

[16] Jan Nouza: Strategies for Developing a Real-Time Continuous Speech Recognition System for Czech Language. In: Text, Speech and Dialogue (eds. Petr Sojka, Ivan Kopeček, Karel Pala) Springer-Verlag, Heidelberg, 2002, pp. 189 – 196, ISBN 3-540-44129-8, ISSN 0302-9743.

[17] Dana Nejedlová: Comparative Study on Bigram Language Models for Spoken Czech Recognition. In: Text, Speech and Dialogue (eds. Petr Sojka, Ivan Kopeček, Karel Pala) Springer-Verlag, Heidelberg, 2002, pp. 197 – 204, ISBN 3-540-44129-8, ISSN 0302-9743.

[18] Mehryar Mohri, Fernando Pereira, Michael Riley: Weighted Finite-State Transducers in Speech Recognition. In: ASR2000, International Workshop on Automatic Speech Recognition: Challenges for the Next Millennium, Paris, 2000.

[19] Jean-Luc Gauvain, Lori Lamel, Gilles Adda, Michèle Jardino: The LIMSI 1998 HUB-4E Transcription System. In: Proceedings of the DARPA Broadcast News Workshop, Herndon, Virginia, 1999, pp. 99 – 104.

[20] Josef Psutka, Pavel Ircing, J. V. Psutka, Vlasta Radová, William J. Byrne, Jan Hajič, Jiří Mírovský, Samuel Gustman: Large Vocabulary ASR for Spontaneous Czech in the MALACH Project. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003. pp. 1821 – 1824, ISSN 1018-4074.

[21] Andrej Žgank, Zdravko Kačič, Bogomir Horvat: Large Vocabulary Continuous Speech Recognizer for Slovenian Language. In: Text, Speech and Dialogue (eds. Václav Matoušek, Pavel Mautner, Roman Mouček, Karel Taušer) Springer-Verlag, Heidelberg, 2001, pp. 242 – 248, ISBN 3-540-42557-8, ISSN 0302-9743.

[22] Tomaž Rotovnik, Mirjam Sepesy Maučec, Bogomir Horvat, Zdravko Kačič: Large Vocabulary Speech Recognition of Slovenian Language Using Data-Driven Morphological Models. In: Text, Speech and Dialogue (eds. Petr Sojka, Ivan Kopeček, Karel Pala) Springer-Verlag, Heidelberg, 2002, pp. 329 – 332, ISBN 3-540-44129-8, ISSN 0302-9743.

[23] Tomaž Rotovnik, Mirjam Sepesy Maučec, Bogomir Horvat, Zdravko Kačič: A Comparison of HTK, ISIP and Julius in Slovenian Large Vocabulary Continuous Speech Recognition. In: Proceedings of ICSLP (eds. John H. L. Hansen and Bryan Pellom), Center for Spoken Language Research, Boulder CO, USA, 2002, pp. 681 – 684, ISBN 1-876346-40-X.

[24] Vassilios Digalakis, Dimitrios Oikonomidis, D. Pratsolis, N. Tsourakis, C. Vosnidis, N. Chatzichrisafis, V. Diakoloukas: Large Vocabulary Continuous Speech Recognition in Greek: Corpus and an Automatic Dictation System. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003. pp. 1565 – 1568, ISSN 1018-4074.

[25] Dimitrios Oikonomidis, Vassilios Digalakis: Stem-based Maximum Entropy Language Models for Inflectional Languages. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003. pp. 2285 – 2288, ISSN 1018-4074.

[26] http://www.speech.sri.com/
visited on August 7, 2005.

[27] Frankie James: Modified Kneser-Ney Smoothing of n-gram Models. RIACS Technical Report 00.07, October 2000.
http://www.riacs.edu/navroot/Research/TR_pdf/TR_00.07.pdf
visited on August 7, 2005.

[28] http://en.wikipedia.org/wiki/Agglutinative_language
cited on August 7, 2005.

[29] Onur Çilingir, Mübeccel Demirekler: A New Decoder Design For Large Vocabulary Turkish Speech Recognition. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003. pp. 1185 – 1188, ISSN 1018-4074.

[30] Kadri Hacioglu, Bryan Pellom, Tolga Ciloglu, Ozlem Ozturk, Mikko Kurimo, Mathias Creutz: On Lexicon Creation for Turkish LVCSR. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003. pp. 1165 – 1168, ISSN 1018-4074.

[31] Máté Szarvas, Sadaoki Furui: Finite-State Transducer Based Hungarian LVCSR with Explicit Modeling of Phonological Changes. In: Proceedings of ICSLP (eds. John H. L. Hansen and Bryan Pellom), Center for Spoken Language Research, Boulder CO, USA, 2002, pp. 1297 – 1300, ISBN 1-876346-40-X.

[32] Vesa Siivola, Mikko Kurimo, Krista Lagus: Large Vocabulary Statistical Language Modeling for Continuous Speech Recognition in Finnish. In: Proceedings of EuroSpeech (eds. Paul Dalsgaard, Børge Lindberg, Henrik Benner, Zheng-Hua Tan), Aalborg University, Denmark, Scandinavia, 2001, pp. 737 – 740, ISBN 87-90834-10-0, ISSN 1018-4074, ISSN 0908-1224.

[33] Vesa Siivola, Teemu Hirsimäki, Mathias Creutz, Mikko Kurimo: Unlimited Vocabulary Speech Recognition Based on Morphs Discovered in an Unsupervised Manner. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003, pp. 2293 – 2296, ISSN 1018-4074.

[34] Tanel Alumäe: Large Vocabulary Continuous Speech Recognition for Estonian Using Morpheme Classes. In: Proceedings of ICSLP (eds. Soon Hyob Kim and Dae Hee Youn), Sunjin Printing Co., 2004, pp. 389 – 392, ISSN 1225-441x.

[35] Akinobu Lee, Tatsuya Kawahara, Kiyohiro Shikano: Julius — an Open Source Real-Time Large Vocabulary Recognition Engine. In: Proceedings of EuroSpeech (eds. Paul Dalsgaard, Børge Lindberg, Henrik Benner, Zheng-Hua Tan), Aalborg University, Denmark, Scandinavia, 2001, pp. 1691 – 1694, ISBN 87-90834-10-0, ISSN 1018-4074, ISSN 0908-1224.

[36] Tom Laureys, Vincent Vandeghinste, Jacques Duchateau: A Hybrid Approach to Compounds in LVCSR. In: Proceedings of ICSLP (eds. John H. L. Hansen and Bryan Pellom), Center for Spoken Language Research, Boulder CO, USA, 2002, pp. 697 – 700, ISBN 1-876346-40-X.

[37] Roeland Ordelman, Arjan van Hessen, Franciska de Jong: Compound Decomposition in Dutch Large Vocabulary Speech Recognition. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003, pp. 225 – 228, ISSN 1018-4074.

[38] Jean-Luc Gauvain, Gilles Adda, Lori Lamel, Martine Adda-Decker: Transcribing Broadcast News: The LIMSI Nov96 Hub4 System. In: Proceedings of the ARPA Speech Recognition Workshop, Chantilly, Virginia, 1997, pp. 56 – 63.

[39] Robert Hecht, Jürgen Riedler, Gerhard Backfried: German Broadcast News Transcription. In: Proceedings of ICSLP (eds. John H. L. Hansen and Bryan Pellom), Center for Spoken Language Research, Boulder CO, USA, 2002, pp. 1753 – 1756, ISBN 1-876346-40-X.

[40] http://www.nist.gov/speech/publications/tw00/html/cts40/cts40.htm visited on August 7, 2005.

[41] Kevin McTait, Martine Adda-Decker: The 300k LIMSI German Broadcast News Transcription System. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003, pp. 213 – 216, ISSN 1018-4074.

[42] Martine Adda-Decker: A Corpus-Based Decompounding Algorithm for German Lexical Modeling in LVCSR. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003, pp. 257 – 260, ISSN 1018-4074.

[43] Long Nguyen, Xuefeng Guo, Richard Schwartz, John Makhoul: Japanese Broadcast News Transcription. In: Proceedings of ICSLP (eds. John H. L. Hansen and Bryan Pellom), Center for Spoken Language Research, Boulder CO, USA, 2002, pp. 1749 – 1752, ISBN 1-876346-40-X.

[44] Toru Imai, Atsushi Matsui, Shinichi Homma, Takeshi Kobayakawa, Kazuo Onoe, Shoei Sato, Akio Ando: Speech Recognition with a Re-Speak Method for Subtitling Live Broadcasts. In: Proceedings of ICSLP (eds. John H. L. Hansen and Bryan Pellom), Center for Spoken Language Research, Boulder CO, USA, 2002, pp. 1757 – 1760, ISBN 1-876346-40-X.

[45] Young-Hee Park, Dong-Hoon Ahn, Minhwa Chung: Morpheme-based Lexical Modeling for Korean Broadcast News Transcription. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003, pp. 1129 – 1132, ISSN 1018-4074.

[46] Dong-Hoon Ahn, Minhwa Chung: Compact Subnetwork-Based Large Vocabulary Continuous Speech Recognition. In: Proceedings of ICSLP (eds. John H. L. Hansen and Bryan Pellom), Center for Spoken Language Research, Boulder CO, USA, 2002, pp. 725 – 728, ISBN 1-876346-40-X.

[47] Marcello Federico, Nicola Bertoldi: Broadcast News LM Adaptation using Contemporary Texts. In: Proceedings of EuroSpeech (eds. Paul Dalsgaard, Børge Lindberg, Henrik Benner, Zheng-Hua Tan), Aalborg University, Denmark, Scandinavia, 2001, pp. 239 – 242, ISBN 87-90834-10-0, ISSN 1018-4074, ISSN 0908-1224.

[48] Julie Brousseau, Jean-François Beaumont, Gilles Boulianne, Patrick Cardinal, Claude Chapdelaine, Michel Comeau, Frédéric Osterrath, Pierre Ouellet: Automated Closed-Captioning of Live TV Broadcast News in French. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003, pp. 1245 – 1248, ISSN 1018-4074.

[49] Gerhard Backfried, Roser Jaquemot Caldés: Spanish Broadcast News Transcription. In: Proceedings of EuroSpeech, ISCA, Geneva-Switzerland, 2003, pp. 1561 – 1564, ISSN 1018-4074.

[50] Jan Nouza, Jindřich Žďánský, Petr David, Petr Červa, Jan Kolorenč, Dana Nejedlová: Fully Automated System for Czech Spoken Broadcast Transcription with Very Large (300K+) Lexicon. In: Proceedings of Interspeech, Lisbon-Portugal, 2005, ISCA, Bonn, Germany, pp. 1681 – 1684, ISSN 1018-4074.

[51] Dana Nejedlová: Lexicon and Language Model Building for Czech Very-Large-Vocabulary Speech Recognition. In: Speech Processing, 14th Czech-German Workshop, Prague 2004, pp. 82 – 92, ISBN 80-86269-11-6.

[52] Jan Hajič, Eva Hajičová, Petr Pajas, Jarmila Panevová, Petr Sgall, Barbora Vidová Hladká: The Prague Dependency Treebank, CDROM LDC2001T10, The Linguistic Data Consortium, Univ. of Pennsylvania, Philadelphia, PA, 2001, ISBN 1-58563-212-0.

[53] Karel Pala, Pavel Rychlý, Pavel Smrž: Text Corpus with Errors. In: Text, Speech and Dialogue (eds. Václav Matoušek, Pavel Mautner), Springer-Verlag, Heidelberg, 2003, pp. 91 – 97, ISBN 3-540-20024-X, ISSN 0302-9743.

[54] Dana Nejedlová: Building a 20K Vocabulary and Language Model for Czech Language. In: Speech Processing, 12th Czech-German Workshop, Prague 2002, pp. 67 – 70, ISBN 80-86269-09-4.

[55] Lingea, s.r.o.: Český korektor pravopisu © 1995-98.

[56] Dana Nejedlová: Building and Evaluation of a Large Vocabulary for a Czech Voice Dictation System. In: ECMS (The 6th International Workshop on Electronics, Control, Measurement and Signals), Liberec, 2003, pp. 74 – 78, ISBN 80-7083-708-X.

[57] Czech National Corpus - SYN2000. Institute of the Czech National Corpus, Faculty of Arts, Charles University, Prague 2000. Available at WWW: <http://ucnk.ff.cuni.cz>.

[58] Zdena Palková: Fonetika a fonologie češtiny. Karolinum, Prague 1997, ISBN 80-7066-843-1.

[59] Christopher D. Manning, Hinrich Shütze: Foundations of Statistical Natural Language Processing. The MIT Press, 1st edition, 1999, ISBN 0-262-13360-1.

[60] Josef Psutka: Komunikace s počítačem mluvenou řečí, Academia, Prague 1995, ISBN 80-200-0203-0.

[61] The International Phonetic Alphabet. Journal of the Phonetic Association, vol. 19, no. 12, Dec. 1989.

[62] Jan Nouza, Josef Psutka, Jan Uhlíř: Phonetic Alphabet for Speech Recognition of Czech. Radio Engineering, vol. 6, no. 4, December 1997, pp. 16 – 20.

[63] Dana Nejedlová: Fonetická transkripce češtiny pomocí třívrstvé neuronové sítě. [Research report]. Liberec 2000, Technical University of Liberec. Faculty of Mechatronics and Interdisciplinary Engineering Studies. http://itakura.kes.tul.cz/kes/public/zprava00.pdf

[64] Marek Volejník: Fonetická transkripce psané a mluvené češtiny pro účely automatického zpracování řeči. [Master's thesis]. Liberec 1999, Technical University of Liberec. Faculty of Mechatronics and Interdisciplinary Engineering Studies.

[65] Dana Nejedlová, Jan Nouza: Phonetic Transcription of Czech Language Using a NETtalk-type Neural Network. In: Speech Processing, 10th Czech-German Workshop, Prague 2000, pp. 37 – 40, ISBN 80-86269-05-1.

[66] Terrence J. Sejnowski, Charles R. Rosenberg: NETtalk: a Parallel Network That Learns to Read Aloud. In: Cognitive Science, 14, 1986, pp. 179 – 211.

[67] Terrence J. Sejnowski, Charles R. Rosenberg: Parallel Networks That Learn to Pronounce English Text. In: Complex Systems, 1, 1987, pp. 145 – 168.

[68] Jan Kolorenč: Evolving Phonological Rules Using Grammatical Evolution. In: POSTER 2004 [CD-ROM]. ČVUT FEL, Prague 2004.

[69] Vladimír Mařík, Olga Štěpánková, Jiří Lažanský a kolektiv: Umělá intelligence (4). Academia, Prague 2003, ISBN 80-200-1044-0.

[70] Jan Nouza, Petr Červa, Jindřich Žďánský, Jan Kolorenč, Petr David: Towards Automatic Transcription of Parliament Speech. In: Electronic Speech Signal Processing, Proceedings of the 16th Conference on Electronic Speech Signal Processing joint with the 15th Czech-German Workshop on Speech Processing, Dresden, Germany, September 2005, TUDpress, pp. 237 – 244, ISBN 3-938863-17-X, ISSN 0940-6832.

[71] Jan Nouza, Jindra Drábková: Combining Lexical and Morphological Knowledge in Language Model for Inflectional (Czech) Language. In: Proceedings of ICSLP (eds. John H. L. Hansen and Bryan Pellom), Center for Spoken Language Research, Boulder CO, USA, 2002, pp. 705 – 708, ISBN 1-876346-40-X.

[72] Ian H. Witten and Timothy C. Bell: The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. IEEE Transactions on Information Theory, 37(4), (1991), pp. 1085 – 1094.

[73] Frederick Jelinek and Robert L. Mercer: Interpolated Estimation on Markov Source Parameters from Sparse Data. In: Proceedings of Workshop on Pattern Recognition in Practice (eds. Gelsema, E. S. and Kanal, L. N.), North Holland, Amsterdam, 1980, pp. 381 – 397.

[74] Thomas M. Cover, Joy A. Thomas: Elements of Information Theory. Wiley-Interscience, 1991, ISBN 0-471-06259-6.

[75] S. S. Chen, E. Eide, M. J. F. Gales, R. A. Gopinath, D. Kanvesky, and P. Olsen: Automatic Transcription of Broadcast News. In Speech Communication, 37 (1-2) (2002) pp. 69 – 87. IBM T. J. Watson Research Center, March 7, 2001. http://www.research.ibm.com/people/r/rameshg/chen-bn-spcom2001.pdf visited on December 27, 2005.

[76] An Vandecatseye, Jean-Pierre Martens, Joao Neto, Hugo Meinedo, Carmen-Garcia Mateo, Javier Dieguez, France Mihelic, Janez Zibert, Jan Nouza, Petr David, Matus Pleva, Anton Cizmar, Harris Papageorgiou, Christina Alexandris: The COST278 pan-European Broadcast News Database. In Proceedings of the LREC 2004, Lisbon, Portugal, May 2004, pp. 873 – 876, ISBN 2-9517408-1-6.

[77] Petr Červa, Jan Škoda, Jan Nouza: Building and Annotating Large Speech Databases for Automatic Speech Recognition. In: Proceedings of Radioelektronika 2004, April 2004, Bratislava, Slovak Republic, pp. 386 – 389, ISBN 80-227-2017-8.

[78] Ye-Yi Wang, Alex Acero, and Ciprian Chelba: Is Word Error Rate a Good Indicator for Spoken Language Understanding Accuracy. In: Proceedings of the IEEE Workshop on Automatic Speech Recognition Workshop and Understanding (ASRU). St. Thomas, US Virgin Islands. November 2003, pp. 577 – 582, ISBN 0-7803-7981-0.

List of Publications

Dana Nejedlová, Jindra Drábková, Jan Kolorenč, Jan Nouza: „Lexical, Phonetic, and Grammatical Aspects of Very-Large-Vocabulary Continuous Speech Recognition of Czech Language". Presented at the 16th Conference on Electronic Speech Signal Processing joint with the 15th Czech-German Workshop on Speech Processing of the Institute of Radio Engineering and Electronics of the Academy of Sciences of the Czech Republic, at the Lichtenstein palace in Prague on September 27, 2005. In: Electronic Speech Signal Processing, Proceedings of the 16th Conference on Electronic Speech Signal Processing joint with the 15th Czech-German Workshop on Speech Processing, Dresden, Germany, September 2005, TUDpress, pp. 224 – 231, ISBN 3-938863-17-X, ISSN 0940-6832.

Jan Nouza, Jindřich Žďánský, Petr David, Petr Červa, Jan Kolorenč, Dana Nejedlová: „Fully Automated System for Czech Spoken Broadcast Transcription with Very Large (300K+) Lexicon". In: proceedings of the 9th European Conference on Speech Communication and Technology Interspeech 2005 (CD-ROM), Lisbon, Portugal, 2005, ISCA, Bonn, Germany, pp. 1681 – 1684, ISSN 1018-4074.

Jan Nouza, Dana Nejedlová, Jindřich Žďánský, Jan Kolorenč: "Very Large Vocabulary Speech Recognition System for Automatic Transcription of Czech Broadcast Programs". In: proceedings of the 8th International Conference on Spoken Language Processing ICSLP 2004 (eds. Soon Hyob Kim and Dae Hee Youn) (Set of 4 Volumes and CD-ROM), Jeju Island, Korea, October 2004, Sunjin Printing Co., pp. 409 – 412, ISSN 1225-441x.

Dana Nejedlová: "Lexicon and Language Model Building for Czech Very-Large-Vocabulary Speech Recognition". Presented at the 14th Czech-German Workshop on Speech Processing of the Institute of Radio Engineering and Electronics of the Academy of Sciences of the Czech Republic, at the Charles University in Prague on September 14, 2004. In: Speech Processing, 14th Czech-German Workshop, Prague 2004, pp. 82 – 92, ISBN 80-86269-11-6.

Dana Nejedlová: "Construction of a Dictation System for Czech Physicians". Presented at the 13th Czech-German Workshop on Speech Processing of the Institute of Radio Engineering and Electronics of the Academy of Sciences of the Czech Republic, at the Charles University in Prague on September 16, Prague 2004, pp. 115 – 117, ISBN 80-86269-10-8.

Dana Nejedlová, Jan Nouza: "Building of a Vocabulary for the Automatic Voice-Dictation System". Presented at the 6th International Conference TSD 2003 in České Budějovice on September 9, 2003. In: Text, Speech and Dialogue (eds. Matoušek, V.; Mautner, P.) Springer-Verlag, Heidelberg, 2003, pp. 301-308, ISBN 3-540-20024-X, ISSN 0302-9743.

Dana Nejedlová: "Building and Evaluation of a Large Vocabulary for a Czech Voice Dictation System". Presented at The 6th International Workshop on Electronics, Control, Measurement and Signals – ECMS 2003 on June 3, 2003. In: ECMS 2003, Liberec, June 2003, pp. 74 – 78, ISBN 80-7083-708-X.

Dana Nejedlová: "Building a 20K Vocabulary and Language Model for Czech Language". In: Speech Processing, 12th Czech-German Workshop, Prague 2002, pp. 67 – 70, ISBN 80-86269-09-4.

Dana Nejedlová: "Comparative Study on Bigram Language Models for Spoken Czech Recognition". Presented at the 5th International Conference TSD 2002 in Brno, Czech Republic, on September 9, 2002. In: Text, Speech and Dialogue (eds. Sojka, P., Kopeček, I., Pala, K.) Springer-Verlag, Heidelberg, 2002, pp. 197-204, ISBN 3-540-44129-8, ISSN 0302-9743.

Dana Nejedlová, Jan Nouza: "Language Model Support for Continuous Speech Recognition in Czech Language". Presented at the IASTED International Conference "SPPRA 2002" in Greece, Crete on June 27, 2002. In: Signal Processing, Pattern Recognition, and Application, Anaheim (USA), Calgary (Canada), Zurich (Switzerland) 2002, pp. 541 – 546, ISBN 0-88986-338-5, ISSN 1482-7921.

Jan Nouza, Dana Nejedlová: "Experiments with Read Speech Recognition in Czech". Presented at the 11th Czech-German Workshop on Speech Processing of the Institute of Radio Engineering and Electronics of the Academy of Sciences of the Czech Republic, at the Charles University in Prague on September 18, 2001. In: Speech Processing, 11th Czech-German Workshop, Prague 2001, pp. 46 – 49, ISBN 80-86269-07-8.

Dana Nejedlová, Marek Volejník: „Transkripce psaného českého textu do fonetické podoby". (Phonetic Transcription of Written Czech Text) In: Počítačové zpracování řeči – cíle, problémy, metody a aplikace (Computerized Processing of Speech – Goals, Problems, Methods, and Applications) (symposium), Technical University of Liberec, Liberec 2001, pp. 10 – 22, ISBN 80-7083-551-6.

Dana Nejedlová, Jan Nouza: "Phonetic Transcription of Czech Language Using a NETtalk-type Neural Network". Presented at the 10th Czech-German Workshop on Speech Processing of the Institute of Radio Engineering and Electronics of the Academy of Sciences of the Czech Republic, at the Charles University in Prague on September 20, 2000. In: Speech Processing, 10th Czech-German Workshop, Prague 2000, pp. 37 – 40, ISBN 80-86269-05-1.

Appendix 1

The Example of the Output of the Recognizer

Reference transcription: na místě je i náš reportér zdeněk ?hekrlík?

ASR: 3 na místě je náš reportér Zdeněk Uhlík Rojík 3

Noise according to Table 3

OOV

= D +I +S

Table of Edit Distances

?hekrlík?	8	7	6	5	4	3	2	2	3
zdeněk	7	6	5	4	3	2	1	2	3
reportér	6	5	4	3	2	1	2	3	4
náš	5	4	3	2	1	2	3	4	5
i	4	3	2	1	1	2	3	4	5
je	3	2	1	0	1	2	3	4	5
místě	2	1	0	1	2	3	4	5	6
na	1	0	1	2	3	4	5	6	7
	0	1	2	3	4	5	6	7	8
		na	místě	je	náš	reportér	zdeněk	uhlík	rojík

Table of Editing Operations

?hekrlík?	d	d	d	d	d	d	d	S	I
zdeněk	d	d	d	d	d	d	M	i	i
reportér	d	d	d	d	d	M	i	i	i
náš	d	d	d	d	M	i	i	i	i
i	d	d	d	D	s	i	i	i	i
je	d	d	d	M	i	i	i	i	i
místě	d	d	M	i	i	i	i	i	i
na	d	M	i	i	i	i	i	i	i
	0	i	i	i	i	i	i	i	i
		na	místě	je	náš	reportér	zdeněk	uhlík	rojík

N (Number of words):	8 (in the reference transcription)
M (Matches):	6
D (Deletions):	1
I (Insertions):	1
S (Substitutions):	1
Number of OOV words:	1

Accuracy = 100 * (N – D – I – S) / N = 100 * (8 – 1 – 1 – 1) / 8 = 62.5%

Correctness = 100 * (N – D – S) / N = 100 * (8 – 1 – 1) / 8 = 75.0%

Ing. Dana Nejedlová

Creation of Lexicons and Language Models for Automatic Broadcast News Transcription

Tvorba slovníků a jazykových modelů pro automatický přepis zpravodajských pořadů

Autoreferát disertační práce

Technická univerzita v Liberci
Fakulta mechatroniky a mezioborových inženýrských studií

20 stran
Náklad: 20 výtisků

Liberec, 2006

**Technická
univerzita
v Liberci**

**Fakulta mechatroniky a mezioborových
inženýrských studií**

Creation of Lexicons and Language Models for Automatic Broadcast News Transcription

Tvorba slovníků a jazykových modelů pro automatický přepis zpravodajských pořadů

Autoreferát disertační práce

Liberec, 2006 **Dana Nejedlová**

Creation of Lexicons and Language Models for Automatic Broadcast News Transcription

Tvorba slovníků a jazykových modelů pro automatický přepis zpravodajských pořadů

Autoreferát disertační práce

Autor:	Ing. Dana Nejedlová
Studijní program:	P2612 Elektrotechnika a informatika
Studijní obor:	2612V045 Technická kybernetika
Státní doktorská zkouška:	Liberec, 10. březen, 2004
Pracoviště:	Katedra informatiky Hospodářská fakulta Technická univerzita v Liberci Hálkova 6, 461 17 Liberec
Školitel:	Prof. Ing. Jan Nouza, CSc.

Rozsah disertační práce a jejích příloh

Počet stran:	88
Počet obrázků:	13
Počet tabulek:	26
Počet vzorců:	49
Počet příloh:	1

Abstract

Industrial civilization generates a large amount of audio & video data. At the same time the knowledge of the information content of the resulting data collections is usually very useful. The research teams of the most developed countries have already solved the problem of transcription of audio signal of human speech into text, but in every language some space for further improvements of speech recognition technology still remains.

This thesis deals with all linguistic aspects of the creation of continuous speech recognizer for the Czech language. It describes the process of the preparation of text corpus, vocabulary for the recognizer, phonetic transcription of the words in the vocabulary, language model, the process of tuning of the recognizer's parameters, and the collection of test speech database.

This database and its predecessors were used for testing various bigram language models, the influence of phonetic transcription and text normalization on the accuracy of recognition of broadcast news.

The speech recognition experiments were carried out with the use of the recognizer developed at SpeechLab, the Laboratory of Computer Speech Processing at the Faculty of Mechatronics and Interdisciplinary Engineering Studies of the Technical University of Liberec. This system is also described in this thesis.

Abstrakt

Průmyslová civilizace vytváří veliké množství audiovizuálních dat. Zároveň je zřejmé, že znalost informačního obsahu výsledných kolekcí dat je obvykle velmi užitečná. Výzkumné týmy z nejvyspělejších zemí již vyřešily problém přepisu zvukového signálu lidské řeči na text, ale pro každý jazyk stále zůstává nějaký prostor pro vylepšování technologie rozpoznávání řeči.

Tato práce se zabývá všemi lingvistickými aspekty tvorby rozpoznávače plynulé řeči pro český jazyk. Popisuje proces přípravy textového korpusu, slovníku pro rozpoznávač, fonetickou transkripci slov ve slovníku, jazykový model, proces ladění parametrů rozpoznávače a tvorbu testovací databáze promluv.

Tato databáze a její předchůdkyně byly využity pro testování různých bigramových jazykových modelů, vlivu fonetické transkripce a normalizace textu na přesnost rozpoznávání zpravodajských pořadů.

Experimenty s rozpoznáváním řeči byly prováděny na rozpoznávači vyvinutém v Laboratoři počítačového zpracování řeči SpeechLab Fakulty mechatroniky a mezioborových inženýrských studií Technické univerzity v Liberci. Tento systém je v práci rovněž popsán.

Obsah

1. Úvod

Přepis zpravodajských pořadů do textové podoby je v případě českého jazyka stále realizován výhradně lidmi. V anglicky mluvícím světě je však tato činnost předmětem zájmu již poměrně známého vědeckého oboru snažícího se tento proces co nejvíce automatizovat. Zmíněný vědecký obor se nazývá „Broadcast News Transcription", se známou zkratkou BNT, a je speciálním oborem širšího oboru zvaného „automatické rozpoznávání spojité řeči". Tento obor je zase speciálním oborem širšího oboru zvaného „automatické rozpoznávání řeči". To napovídá, že dosti záleží na tom, zda rozpoznávaná řeč je spojitá.

Opakem řeči spojité je řeč izolovaná. První úspěšné pokusy s rozpoznáváním izolované řeči, které spočívaly v tom, že člověk říká jednotlivá slova oddělená pauzou do počítače a počítač jemu známá slova zapisuje, byly realizovány v Bellových laboratořích v USA v 50. letech 20. století. Je pozoruhodné, že úspěšné pokusy s rozpoznáváním spojité řeči, kdy počítač umí sám rozpoznávanou promluvu rozdělit na jednotlivá slova, byly realizovány až přibližně o 20 let později.

Rozpoznávání spojité řeči je totiž o hodně složitější než rozpoznávání jednotlivě vyslovovaných slov. Pro rozpoznávání izolovaných slov bylo potřeba vyřešit tři následující úkoly:

1. Počítačová reprezentace zvuku.
2. Extrakce příznaků lidské řeči. Zvukový záznam s lidskou řečí obsahuje mnohem více informací, než je nutné mít pro rozpoznání řeči. Vybrání pouze těch parametrů, které popisují řeč umožní vytvořit dobré modely slov, které má rozpoznávač umět rozpoznat.
3. Algoritmizace porovnání posloupnosti příznaků v rozpoznávaném signálu s modely slov.

Pro rozpoznávání spojité řeči bylo potřeba vyřešit úkolů mnohem více:

1. Počítačová reprezentace zvuku.
2. Segmentace spojitého vstupního signálu na úseky dlouhé několik slov.
3. Extrakce příznaků lidské řeči.
4. Reprezentace slov pomocí takzvaných skrytých markovských modelů a vývoj algoritmů pro práci s nimi.
5. Segmentace akustického signálu slov na fonémy, čímž se získají data pro trénování skrytých markovských modelů.
6. Trénování skrytých markovských modelů reprezentujících fonémy.
7. Fonetické transkripce textové podoby slov.
8. Jazykové modelování.
9. Zpracování velkých textových korpusů.
10. Tvorba slovníku pro rozpoznávač.
11. Tvorba a ladění rozpoznávače.

Minimálně body 7 a 8 jsou závislé na jazyku, takže je zde prostor pro originální objevy laboratoří pracujících s různými jazyky.

Úspěšná realizace automatického rozpoznání spojité řeči bude značnou pomocí pro velký úkol dneška, kterým je získávání znalostí z multimediálních dat, která denně vznikají.

2. Systém pro automatický přepis zpravodajských pořadů vyvinutý v laboratoři SpeechLab

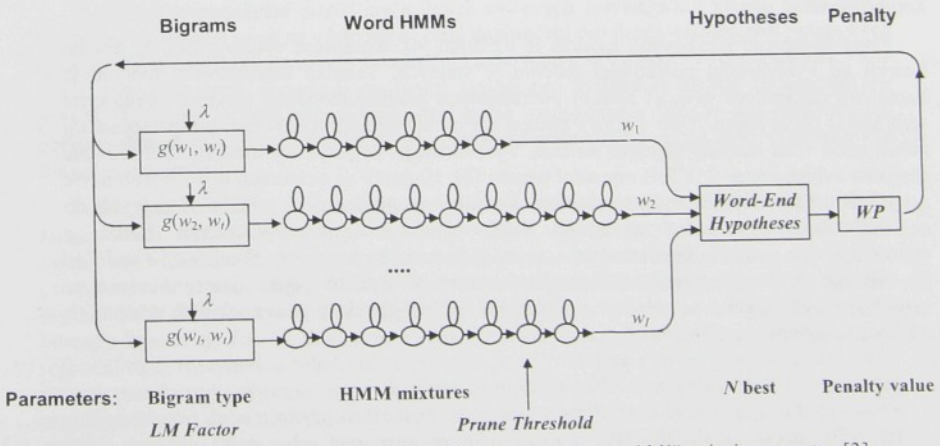
Výzkum popsaný v této práci byl proveden v české laboratoři pro zpracování řeči zvané SpeechLab. Tato laboratoř byla založena profesorem Janem Nouzou v roce 1994 a je součástí Fakulty mechatroniky Technické univerzity v Liberci.

Výzkum ve SpeechLabu začal rozpoznáváním izolovaných slov a krátkých frází. První systém pro rozpoznávání spojité řeči byl zde vyvinut v roce 1999 a jeho slovník čítal 200 slov. Tento systém byl postupně zdokonalován, aby pracoval rychleji a s většími slovníky, takže v roce 2005 rozpoznávač této laboratoře dokáže pracovat se slovníkem o 312 tisících slovech.

Úkolem rozpoznávače je najít takovou sekvenci slov, která maximalizuje pravděpodobnost (1).

$$P(w_1^*,...,w_N^*) = \underset{w,N}{Max} \sum_{n=1}^{N} (\lambda \cdot \ln g(w_{n-1}, w_n) + WP + \ln V(w_n)) \qquad (1)$$

Člen $g(w_{n-1}, w_n)$ v rovnici (1) reprezentuje jazykový model neboli bigramovou pravděpodobnost, že slovo w_n následuje za slovem w_{n-1}. Člen $V(w_n)$ v rovnici (1) je roven pravděpodobnost, že slovo ... akustickému skóre slova w_n dosaženého jeho skrytým markovským modelem vyhodnoceným na sekvenci vektoru příznaků vypočtených ze vstupního signálu. Z důvodu zcela rozdílné povahy jazykového modelu $g(w_{n-1}, w_n)$ a akustického skóre $V(w_n)$ má člen pro jazykový model váhu zvanou λ nebo *LM Factor*. Některá naše i cizí pozorování ukazují, že rozpoznávač má tendenci preferovat kratší slova před delšími. Pro potlačení tohoto jevu jsme zavedli parametr zvaný *Word Insertion Penalty* nebo *WP*, který zhoršuje skóre každého posuzovaného slova. Rovnice (1) je řešena pomocí takzvaného Viterbiho algoritmu popsaného v literatuře [1] a [2] (str. 176 – 180) obohaceného o techniky prořezávání hypotéz. Schéma rozpoznávače ukazuje Obrázek č. 1.



Obrázek č. 1 Struktura rozpoznávání sekvence slov se všemi klíčovými parametry [3]

Vstupní signál je vzorkován pomocí 16 bitů s frekvencí 8 kHz a jsou z něj extrahovány 39-příznakové MFCC vektory pro každý frame. MFCC je zkratka pro „mel frequency cepstral coefficients“, česky „kepstrální koeficienty mel-frekvence“ vysvětlované například v [1] (str. 75 – 87) Všechna slova ve slovníku rozpoznávače jsou reprezentována zřetězenými modely fonémů. Jednotlivé modely fonémů korespondují s třístavovými skrytými markovskými modely trénovanými na několikahodinové anotované databázi promluv. Skryté markovské modely jsou 32-mixturové monofóny.

Akustické modely všech slov jsou vyhodnocovány paralelně s předpokladem, že libovolné slovo může začít v libovolném framu. Frame je 10-milisekundový segment akustického vstupního signálu. Každý frame je převeden do akustického příznakového vektoru během procesu zvaného parametrizace. V každém framu jsou méně pravděpodobné cesty ve Viterbiho algoritmu odříznuty, aby se urychlil výpočet. To je ovlivňováno parametrem zvaným *Prune Threshold*. Slova, jejichž skóre vypočtené Viterbiho algoritmem dělené nejlepším dosud nalezeným skóre je nižší než *Prune Threshold*, jsou dočasně odstraněny z výpočtů. Nejvyšší skóre v konečných stavech skrytých markovských modelů patří nejpravděpodobnějším slovům, která končí v daném framu. Pro další výpočty je vybrán jen omezený počet nejlepších slovních kandidátů. Tento počet slov je určen parametrem zvaným *Number of Word-End Hypotheses*. Skóre každého slovního kandidáta je penalizováno konstantou zvanou *WP* nebo *Word Insertion Penalty*. V dalších výpočtech jsou jako následující slovo brány v úvahu hypotézy o všech možných slovech ve slovníku rozpoznávače. Počáteční skóre nového slova je rovno skóre slova, které bylo předtím ukončeno a ke kterému je připočtena bigramová pravděpodobnost $\lambda \cdot \ln g(w_{n-1}, w_n)$ nového slova za podmínky, že mu předchází předtím ukončené slovo.

Ta část výše popsaného rozpoznávače, která má nejblíže tématu této práce, je bigramový jazykový model. Když má slovník rozpoznávače 312 000 slov, bigramový jazykový model je tabulka pravděpodobností posloupností všech možných slovních párů z tohoto slovníku, takže tato tabulka má počet hodnot rovný druhé mocnině čísla 312 000. Když je každá hodnota v počítači reprezentována 4 byty, celý jazykový model zabírá 363 GB paměti. Vzhledem k hardwarovým omezením současné výpočetní techniky je nezbytné tuto tabulku komprimovat.

Popis komprese jazykového modelu je v článku [4]. Komprese využívá fakt, že mnoho hodnot ve vyhlazeném jazykovém modelu je stejných. Tabulka bigramového modelu je rozdělena do vektorů $\mathbf{h}(w_{n-1})$ hodnot podmíněných pravděpodobností slovních párů, které sdílí stejné první slovo. Tyto vektory mohou být efektivně komprimovány, protože obsahují menší nebo větší skupiny stejných hodnot. Výsledkem je, že jazykový model 312 tisícového slovníku zabírá pouze 251 MB operační paměti [5]. Hodnoty ve vektorech $\mathbf{h}(w_{n-1})$ jsou navíc uspořádány nikoliv podle svého pořadí ve slovníku ale podle svých hodnot od nejvyšší do nejnižší. Toto uspořádání umožňuje další výpočetní úsporu. V každém framu je vyhodnocováno pouze tolik vektorů $\mathbf{h}(w_{n-1})$, kolik je hodnota parametru *Number of Word-End Hypotheses*. A z tohoto malého množství vektorů je využito pouze omezené množství nejvyšších hodnot, protože méně pravděpodobná následující slova jsou z výpočtů odstraněna díky parametru *Prune Threshold*.

3. Stav výzkumu automatického přepisu zpravodajských pořadů ve světě

Výzkum různých laboratoří se liší hlavně dvěma faktory: jazykem řeči, která je automaticky rozpoznávána, a tím, zda-li je používán vlastní nebo veřejně dostupný rozpoznávač.

Hlavní vlastnost jazyka, která ovlivňuje úspěšnost rozpoznávání, je takzvané pokrytí textu slovníkem. Rozpoznávání v jazycích, které s určitým počtem nejfrekventovanějších slov pokryjí co největší procento slov v předem neznámém textu, je ve velké výhodě. Takovým jazykem je angličtina. Když se k tomu ještě přičte fakt, že angličtina je jazykem zemí s nejrozvinutějším vědeckým výzkumem, není divu, že rozpoznávání řeči v anglickém jazyce má již mnoho úspěšných komerčních aplikací. Výsledky výzkumu rozpoznávání angličtiny jsou využívány i v laboratořích zabývajících se rozpoznáváním jiných jazyků, ale rozdílnost jazyků často vede k tomu, že pro jiné jazyky je nutné zvolit jiné metody.

Jedním z nejvážnějších problémů způsobených odlišností angličtiny od ostatních jazyků je množství slov nutné k dosažení určitého pokrytí. Zatímco slovník 60 tisíc nejfrekventovanějších anglických slov pokryje 99 % textu, v češtině je to 92 %, jak uvádí práce [6] na straně 66. Výsledkem výzkumu rozpoznávání angličtiny je i několik rozpoznávačů, které jsou veřejně dostupné. Některé výzkumné týmy staví svůj výzkum na nich, protože vývoj vlastního rozpoznávače je velmi náročný. Pokud je předmětem zájmu těchto týmů jazyk s mnohem menším pokrytím, než má angličtina, musí tyto týmy překonávat mnohé možná nepřekonatelné problémy, protože velikost slovníku veřejně dostupných rozpoznávačů je dimenzována pouze pro potřeby angličtiny na přibližně 60 tisíc slov.

Výsledkem je však mnoho originálních přístupů k řešení, které byly vymýšleny při rozpoznávání jiných jazyků, než je angličtina. Mezi tyto přístupy patří například rozdělování nebo naopak spojování lexikálních jednotek rozpoznávače, různé způsoby vyhlazování jazykového modelu a jeho přizpůsobování tématu promluv nebo také používání jazykového modelu založeného na slovních gramatických kategoriích.

Výzkumy rozpoznávání angličtiny a jiných světových jazyků, které nemají příliš vážné problémy s pokrytím, mají za výsledek některá doporučení pro úspěšnou realizaci praktických aplikací. Je to například stanovení maximální chybovosti a doby odezvy rozpoznávače, která nesmí být překročena, chceme-li realizovat systém pro titulkování živých zpravodajských pořadů. Zajímavá je také metoda vylepšení práce rozpoznávače tím, že mu zpravodajský pořad přeříkává speciální osoba, která má znalosti jeho slovníku a ví, jak mají vypadat dobře formulované titulky.

Porovnáme-li češtinu s ostatními jazyky, které jsou předmětem zájmu automatického rozpoznávání řeči, vidíme, že čeština jako vysoce ohebný jazyk má největší problémy s pokrytím. Chceme-li někdy dosáhnout úspěšnosti rozpoznávání češtiny srovnatelné s rozpoznáváním angličtiny s 60-tisícovým slovníkem, musíme použít slovník o několika stech tisících slovních tvarů. Práce [6] řeší tento problém tím, že slova rozkládá na jejich kmeny a koncovky. Tak lze dosáhnout vyššího pokrytí se stejným slovníkem i vyšší přesnosti rozpoznávání. Nevýhodou tohoto přístupu však je, že rozložení slov na části zkracuje úseky textu popsané jazykovým modelem. Trigramový model z rozložených slov nemůže popsat ani pravděpodobnost dvou následujících celých slov, pouze buďto začátek slova, jeho koncovku a začátek druhého slova nebo koncovku prvního slova a celé následující slovo. Hlavním výsledkem práce [6] je zjištění, že pro dosažení úspěšnosti rozpoznávání češtiny srovnatelné s angličtinou bude patrně nutné zvětšit velikost slovníku rozpoznávače.

4. Cíle disertační práce

Výzkum popsaný v této práci je založen na předpokladu, že slova jsou přirozené lingvistické jednotky nesoucí sémantickou, syntaktickou a gramatickou informaci zakódovanou do posloupnosti fonémů. Všechny tyto čtyři vlastnosti (sémantická, syntaktická, morfologická a fonologická) spolu úzce souvisí a mohou být reprezentovány lépe na úrovni celých slov než na úrovni slovních částí (morfémů) nebo slovních gramatických kategorií [7]. Abychom dosáhli dobrých výsledků pomocí tohoto způsobu reprezentace, musíme splnit tyto cíle:

1. Příprava textového korpusu
2. Sestavení slovníku obsahujícího několik stovek tisíců slov
3. Fonetická transkripce slov ve slovníku
4. Výpočet různých bigramových jazykových modelů
5. Příprava testovací databáze promluv
6. Testování slovníku, jazykového modelu a parametrů rozpoznávače na databázi promluv
7. Vytvoření kritérií pro měření kvality přepisu

5. Řešení

5.1. Textový korpus

Účelem textového korpusu je získání informace o frekvencích jednotlivých slov a jejich řetězců. Ze seznamu nejfrekventovanějších slov se sestavuje slovník rozpoznávače. Z frekvencí řetězců slov se počítá *n*-gramový jazykový model.

Korpus si sbírají členové SpeechLabu sami. Hlavním zdrojem korpusu jsou různé internetové noviny. V roce 2005 měl korpus velikost 2,6 GB prostého (neformátovaného) textu a stále se zvětšuje.

Původní formát textu je HTML, takže první operace, kterou je třeba udělat, je odstranění HTML značek. Výsledný prostý text se musí vyčistit. Čištění korpusu lze shrnout do následujících bodů [5]:

1. Každá věta je ve finálním korpusu umístěna na jeden řádek. Identifikace vět je automatická a algoritmus, který byl pro tento účel vyvinut obsahuje mnoho pravidel říkajících, která tečka skutečně označuje konec věty.
2. Jednotlivá slova v závorkách jsou potom vymazána. Taková slova jsou totiž většinou z hlediska jazykového modelování nezajímavé zkratky.
3. Opakující se záhlaví, zápatí a formátovací znaky jsou vymazány.
4. Nesklonné zkratky jsou přepsány na celá slova.
5. Výrazy typu *x-letý*, kde *x* je psáno číslicemi, jsou přepsány na celá slova.
6. Každé slovo je převedeno na malá písmenka a každý interpunkční znak je obklopen mezerou, aby bylo možné je počítat a provádět analýzu jejich sousedů.
7. Čísla znamenající hodiny a datumy a některá další čísla jsou přepsány do jejich mluvené podoby. Řadové číslovky, před kterými je předložka, jsou přepsány do jejich gramaticky správného pádu a rodu s pomocí českého morfologického analyzátoru.

8. Slova jsou přepsána do jejich standardní ortografické podoby. Protože mnoho slov, obzvláště těch cizího původu, má alternativní způsoby zápisu, sjednotili jsme jejich ortografii na nejčastější varianty. To trochu zmenšilo slovník rozpoznávače a zvětšilo jeho pokrytí normalizovaného textu. Ručně jsme nalezli 35 000 slovních tvarů, které by měly být přepsány. Tato přepisovací pravidla jsou také aplikována na referenční transkripce promluv určených pro testování rozpoznávače. [7] Tento proces se obvykle nazývá normalizace ortografie.
9. Kolokace, neboli fráze slov, které se často vyskytují vedle sebe, jsou spojeny speciálním znakem, aby s nimi bylo při počítání jazykového modelu zacházeno jako s jedním slovem. V současné době máme v našem slovníku 1 700 kolokací.

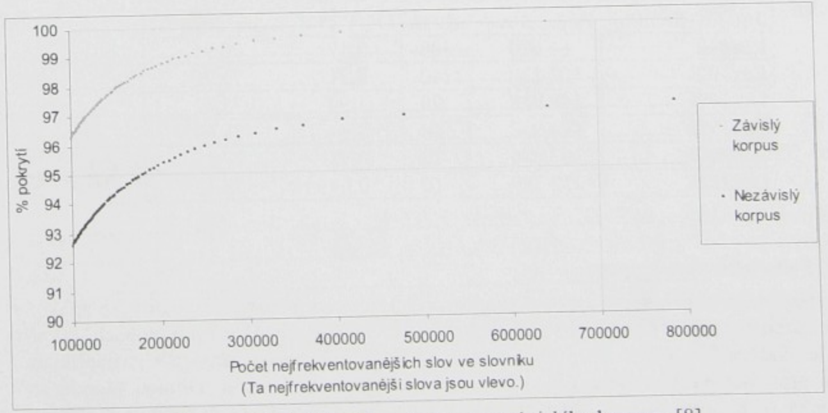
5.2. Slovník

5.2.1. Výběr slov

Spolu s tím, jak jsme zvětšovali slovník našeho rozpoznávače, zabývali jsme se i metodami výběru slov do slovníku a teoretickým odhadem pokrytí velkých slovníků.

Nejdříve jsme porovnali metody výběru slov pomocí statistické analýzy korpusu a pomocí generování slovních tvarů z databáze slovních kmenů a koncovek. Zjistili jsme, že analýza korpusu je sice pracná, protože mnoho slov musí kontrolovat člověk, ale jejím výsledkem je slovník, který má výrazně vyšší pokrytí nezávislého textu než slovník vzniklý generováním slovních tvarů.

Sloučením množin slov vzniklých oběma metodami výběru vznikl poměrně dobře anotovaný slovník 800 tisíc slovních tvarů. Tento slovník jsme využili k analýze pokrytí. Výsledek této analýzy nám pomáhá odhadnout, jak velký slovník máme pro rozpoznávač sestavit, chceme-li, aby měl určité pokrytí. Obrázek č. 2 ukazuje křivku pokrytí pro slovníky s počtem nejfrekventovanějších slov mezi 100 000 a 800 000.



Obrázek č. 2 Pokrytí závislého a nezávislého korpusu [8]

Pokud v angličtině je možné, aby 60-tisícový slovník pokryl 99 % textu, jak se uvádí v kapitole č. 3, tak v češtině ani 800-tisícový slovník takového pokrytí nedosahuje.

Analýza pokrytí nás utvrdila v tom, že je nutné stále rozšiřovat slovník rozpoznávače. Nejnovější verze slovníku pro rozpoznávač spojité řeči z roku 2005 má 312 tisíc slov.

5.2.2. Slučování a rozdělování slov

Článek [7] uvádí, že když do slovníku o 310 tisících slovech přidáme 1 708 nových slov tvořených zřetězeními slov, která ve slovníku už jsou, stoupne přesnost rozpoznání ze 75,5 % na 78,2 %. Tato zřetězení se odborně nazývají kolokace a musí být podchyceny již ve fázi čištění korpusu, viz bod 9 v kapitole č. 5.1.

Udělali jsme také pokus zjišťující, zda-li naopak rozdělení některých slov také nezlepší rozpoznávání. Motivací k tomu byla úvaha, že některá česká slova mají velmi časté předpony *ne-*, *nej-* a *nejne-* a mnoho přídavných jmen obsahuje číslovky, například *dvacetiletý*. Rozložením těchto slov by byl zredukován slovník, který by současně dosahoval vyššího pokrytí nezávislých textů. Výsledkem je zjištění, že dekompozice slov rozpoznání většinou nezlepšuje. Souhrnná přesnost rozpoznávání na databázi televizních i rozhlasových zpráv popsaných v kapitole č. 5.5 je 80,45 % s celými slovy a 79,76 % s dekomponovanými slovy i když procento slov rozpoznávaného testu nenalezených ve slovníku rozpoznávače kleslo z 1,35 % na 1,30 %, když byla slova rozložena (dekomponována).

5.2.3. Závislost přesnosti rozpoznávání na velikosti slovníku

Tabulka č. 1 ukazuje závislost přesnosti rozpoznávání (Accuracy) na počtu slov ve slovníku. „Rank" je prahová četnost slov v trénovacím korpusu, kterou musela všechna slova v daném slovníku přesáhnout. „OOV" je procento slov v rozpoznávaném textu, která nebyla nalezena ve slovníku. Všechny menší slovníky byly podmnožinou slovníku „Lex312k". Testovací databáze promluv jsou 3 kompletní televizní zpravodajské pořady, viz kapitola č. 5.5.

Tabulka č. 1 Rozpoznávací skóre dosažená se slovníky různých velikostí [5]

Jméno slovníku	Počet slov	Rank	OOV [%]	Accuracy [%]
Lex64k	64 620	300	5,17	70,96
Lex102k	102 228	140	3,31	73,75
Lex149k	148 928	70	1,94	75,62
Lex195k	194 932	40	1,34	76,64
Lex257k	257 086	20	0,97	77,27
Lex312k	312 289	10	0,64	78,13

5.3. Fonetická transkripce

Fonetická transkripce vyjadřuje posloupnost zvuků mluvené řeči řetězci textových znaků. V oboru komunikace lidí s počítači se fonetická transkripce uplatňuje ve dvou úlohách. První úlohou je syntéza řeči podle textového vstupu. Možnou aplikací syntézy řeči je například software pro počítače pro slepé lidi a informační služby po telefonu. Druhou úlohou je automatické rozpoznání řeči. V obou případech musí být znaky textu psané formy jazyka (grafémy) namapovány na znaky zastupující zvuky vyslovované při čtení tohoto textu (fonémy). Fonetická transkripce slouží v obou úlohách jako spojka mezi textovou formou a akustickými modely slov.

Při řečové syntéze i rozpoznávání je fonetická transkripce užita ve dvou *fázích*. Nejdříve se musí trénovací databáze promluv přepsat na text a tento text musí být přepsán foneticky. Výsledek je využit v procesu segmentace. Při segmentaci se zvukový signál s řečí dělí na segmenty obsahující jednotlivé fonémy. Tyto segmenty jsou potom využity pro trénování akustických modelů fonémů. Počáteční fáze segmentace se musí provádět ručně. Když jsou natrénovány spolehlivější akustické modely, segmentace může být více či méně automatická. Ve druhé fázi je fonetická transkripce aplikována na text, který má vyslovovat počítač, a v případě automatického rozpoznávání řeči se fonetická transkripce využívá k přepisu slov ve slovníku rozpoznávače.

Fonetickou transkripci SpeechLab realizuje pomocí vlastních počítačových programů. První takový program byl v této laboratoři vyvinut v roce 1999. Používal množinu přepisovacích pravidel navrženou pro český jazyk v knížce [9] (str. 101 – 106). Pomocí této množiny pravidel však nebylo přepsáno správě dostatečně velké množství slov. Základní pravidla musíme stále rozšiřovat o výjimky. Ohebnost českého jazyka množství těchto výjimek velmi zvětšuje.

Zkusili jsme také realizovat fonetickou transkripci pomocí neuronové sítě. Naše neuronová síť četla 5 znaků přepisovaného textu a odhadovala foném, na který se má přepsat prostřední znak z těchto 5 znaků. Síť se to nejdříve učila na anotovaném seznamu slov a jejich fonetických transkripcí. Naučená síť přepisovala nová slova. Výsledky dosažené s touto neuronovou sítí byly horší než se systémem založeným na pravidlech. Zatímco systém založený na pravidlech je možné zdokonalovat vylepšováním množiny pravidel, neuronová síť má některá principiální omezení, která například brání, aby se naučila výjimky.

V roce 2004 jsme použili genetické algoritmy pro automatizovaný sběr přepisovacích pravidel pro náš původní systém vyvíjený od roku 1999. V genetickém algoritmu se náhodně modifikují přepisovací pravidla a testují se na anotovaném seznamu slov a jejich fonetických transkripcí. Výstupem jsou pravidla, která úspěšně přepisují slova v seznamu a žádná nezkazí. Takto se podařilo obohatit náš systém pro fonetickou transkripci o nová pravidla.

Náš současný systém pro fonetickou transkripci dává každému slovu jen jedinou variantu výslovnosti. Mnoho slov má však z různým důvodů více variant. Dodatečné varianty zatím ve slovníku rozpoznávače doplňujeme ručně a jejich vliv na přesnost rozpoznávání je značný.

5.4. Jazykový model

5.4.1. Účel a podoba jazykového modelu

Jazykový model reprezentuje znalost o jazyce, která obvykle říká, jak jsou určitá slova pravděpodobná v určitém kontextu.

Ve většině rozpoznávačů spojité řeči jsou využívány takzvané *n*-gramové jazykové modely. *N*-gramový jazykový model je tabulka podmíněných pravděpodobností slova za podmínky, že určitá posloupnost $n - 1$ slov mu v promluvě předcházela. Tyto pravděpodobnosti jsou počítány z velkého trénovacího korpusu. Díky tomu jsou *n*-gramové jazykové modely také nazývány statistické. Při výpočtu jazykového modelu jsou z trénovacího korpusu vybírána pouze slova patřící do slovníku rozpoznávače. Náš rozpoznávač používá bigramové jazykové modely, které jsou speciálním případem *n*-gramových jazykových modelů pro $n = 2$.

5.4.2. Výhody *n*-gramových jazykových modelů

1. *N*-gramové jazykové modely jsou dostatečně flexibilní, aby umožnily rozpoznávání libovolných promluv.
2. Algoritmus výpočtu *n*-gramové statistiky je nezávislý na jazyku.

3. Algoritmus výpočtu a používání *n*-gramového jazykového modelu je relativně jednoduchý, obzvláště v případě velmi malých slovníků.
4. *N*-gramové jazykové modely jsou levo-pravé. To znamená, že předpovídají budoucnost z minulosti. Z toho důvodu jsou dobře integrovatelné s akustickými modely založenými na skrytých markovských modelech, které jsou také levo-pravé.
5. *N*-gramové jazykové modely mohou být snadno přizpůsobeny určité tématické doméně. *N*-gramy patřící do té části korpusu, která se týká požadovaného tématu mohou dostat větší váhu než ostatní *n*-gramy a výsledné dvě skupiny *n*-gramů se mohou snadno sloučit. Výsledky této techniky jsou často překvapivě dobré.
6. *N*-gramové jazykové modely popisující pravděpodobnosti slovních tvarů mohou být snadno kombinovány s *n*-gramovými jazykovými modely popisujícími pravděpodobnosti gramatických kategorií slovních tvarů.

5.4.3. Nevýhody *n*-gramových jazykových modelů

1. *N*-gramové jazykové modely se skládají z velkého počtu parametrů rovného m^n, kde *m* je velikost slovníku a *n* je řád *n*-gramového jazykového modelu. Velikost slovníku *m* musí být obvykle veliká, což znamená, že *n* musí být obvykle menší než 4.
2. Prakticky použitelné *n*-gramové jazykové modely dokáží popsat jen lokální závislosti slov, ale reálné závislosti v jazyce často sahají dále než přes dvě nebo tři slova.
3. *N*-gramové jazykové modely by měly obsahovat pravděpodobnosti všech možných slovních dvojic nebo trojic. Ale žádný trénovací korpus není tak rozsáhlý, aby obsahoval každou možnou posloupnost dvou nebo tří slov. Například náš 2,6-GB trénovací textový korpus obsahuje 60 228 569 různých slovních párů složených ze slov v našem 312-tisícovém slovníku. Počet slovních párů, jejichž pravděpodobnost musí být v jazykovém modelu určena je ve skutečnosti $312\,000^2$. To znamená, že náš korpus obsahuje pouze 0,06 % všech možných slovních párů. Některé slovní páry, které chybí v korpusu by skutečně měly mít téměř nulovou pravděpodobnost. Některé jiné chybějící slovní páry jsou však z hlediska gramatiky daného jazyka přípustné. Umění jazykového modelování spočívá v rozlišení té jedné skupiny chybějících slovních párů od druhé a určení správných pravděpodobností pro přípustné slovní páry. Tuto úlohu řešíme takzvaným vyhlazováním jazykových modelů popsaným v kapitole č. 5.4.4.
4. *N*-gramové jazykové modely musí být vypočteny z velmi rozsáhlých trénovacích korpusů. Příprava takových korpusů je velmi pracná (viz kapitola č. 5.1), a výpočet frekvencí slovních posloupností musí být realizován nějakým efektivním způsobem.
5. Bigramové jazykové modely pro slovníky přesahující přibližně 10 tisíc slov musí být reprezentovány způsobem, který umožňuje jak jejich vměstnání do operační paměti současných osobních počítačů tak i efektivní využívání informací, které obsahují, během rozpoznávání řeči. Naše řešení tohoto problému je naznačeno na konci kapitoly č. 2.

K těmto nevýhodám *n*-gramových jazykových modelů platících obecně pro každý jazyk je nutno přičíst specifické vlastnosti českého jazyka.

Za prvé, díky ohebnosti češtiny je nutné mít ve slovníku rozpoznávače řádově sta tisíce položek, což znamená, že nemůžeme použít jazykový model vyššího řádu než bigramový.

Za druhé, v češtině je mnoho pravidel o shodě mezi rodem, číslem a pádem, takže velké množství slovních párů se nikdy v gramatické české větě neobjeví. To má za následek velmi řídkou matici četností slovních párů nalezených v trénovacím korpusu. Tato matice musí být nějak vyhlazena (viz kapitola č. 5.4.4), ale vyhlazení, které nebere v úvahu informaci o české gramatické shodě dává příliš velikou pravděpodobnost bigramům, které mají zůstat nulové.

Za třetí, vedlejším efektem pravidel gramatické shody je relativně volný pořádek slov v české větě. Mluvnická shoda nese informaci o vztazích podmětů, předmětů a přísudků, takže není nutné určitým způsobem slova ve větě řadit. To sice redukuje řídkost jazykového modelu, ale jeho prediktivní schopnost se tím spíše zhoršuje.

5.4.4. Vyhlazování jazykových modelů

Vyhlazování nahrazuje nulové pravděpodobnosti v jazykovém modelu nenulovými pravděpodobnostmi. I když je takto dána nenulová pravděpodobnost slovním vazbám, které nejsou gramaticky povoleny, máme experimentálně ověřeno, že přesnost rozpoznávání se vyhlazením jazykového modelu výrazně zvýší.

Pro vyhlazování používáme metodu zvanou Witten-Bell discounting, viz vzorce (2) a (3), kde $C(w_1, w_2)$ je četnost dvojice slov w_1, w_2 v trénovacím korpusu, $C(w_1)$ je četnost slova w_1, $T(w_1)$ je počet druhů slov, které se v trénovacím korpusu objevily za slovem w_1, a V je velikost slovníku neboli počet všech možných druhů slov.

$$P(w_2|w_1) = \frac{C(w_1, w_2)}{C(w_1) + T(w_1)} \quad \text{když } C(w_1, w_2) > 0 \tag{2}$$

$$P(w_2|w_1) = \frac{T(w_1)}{(V - T(w_1)) \cdot (C(w_1) + T(w_1))} \quad \text{když } C(w_1, w_2) = 0 \tag{3}$$

K této metodě jsme přidali dvě pravidla vyjádřená vzorci (4) a (5). Ověřili jsme pomocí výpočtu křížové perplexity jazykového modelu proti textu z testovací databáze, že takto upravené vyhlazování lépe vystihuje předem neznámá data než původní forma Witten-Bell discounting (2) a (3) nebo Witten-Bell discounting kombinované s vyhlazováním, které zvýší četnost všech možných dvojic slov o jednu (takzvané add-one smoothing).

$$P(w_2|w_1) = \frac{C(w_1, w_2) \cdot (C(w_1) + 2T(w_1) - V)}{C(w_1) \cdot (C(w_1) + T(w_1))} \quad \text{když } 2T(w_1) > V \text{ a } C(w_1, w_2) > 0 \tag{4}$$

$$P(w_2|w_1) = \frac{1}{C(w_1) + T(w_1)} \quad \text{když } 2T(w_1) > V \text{ a } C(w_1, w_2) = 0 \tag{5}$$

Některé jiné vyhlazovací metody by mohly vést i k lepším výsledkům. Nepoužíváme je proto, že jejich počítačová reprezentace by zpomalila proces rozpoznávání. Souvisí to se způsobem, jak reprezentujeme jazykový model, viz konec kapitoly č. 2.

5.5. Testovací databáze promluv

Výsledky publikované v této práci byly pořízeny na třech testovacích databázích promluv.

První databáze obsahuje 1 600 vět namluvených neprofesionálními mluvčími, kteří četli texty z novin do mikrofonu. Tato databáze má celkem 16 027 slov. Její popis spolu s výsledky byl publikován v článcích [4] a [10].

Druhá databáze je tvořena třemi kompletními zpravodajskými televizními pořady. Tato databáze má celkem 8 451 slov. Její popis spolu s výsledky byl publikován v článku [11].

Třetí databáze byla vytvořena pro tuto práci a bude využívána i v budoucnu. Je tvořena různými rozhlasovými zpravodajskými pořady, je v ní řečeno 13 081 slov 61 mluvčími během

1,5 hodiny. Výsledek základního rozpoznávacího experimentu na této databázi ukazuje Tabulka č. 2.

Tabulka č. 2 Analytický pohled na přesnost (Accuracy) v procentech základního rozpoznávacího experimentu na testovací databázi rozhlasových zpráv (Všechna slova byla převedena na malá písmena. Jazykový model byl z celých slov.)

Rozhlasová stanice	**Pohlaví**	**Styl promluvy**	**Zvukové podmínky**			**Celkem**	**Celkem**
			Čisté	**Nízká věrnost (např. telefon)**	**Hluk na pozadí (řeč, hudba)**		
Radiožurnál	muž	profesionál	87,851	81,053	77,635	84,730	82,827
Radiožurnál	muž	host	76,739	54,869	78,689	69,109	
Radiožurnál	žena	profesionál	90,709	83,099	83,668	89,768	
Radiožurnál	žena	host	83,333	69,748	8,571	58,721	
BBC	muž	profesionál	83,006	.	75,749	81,850	80,589
BBC	muž	host	73,036	67,130	78,298	70,043	
BBC	žena	profesionál	87,681	.	80,465	87,077	
BBC	žena	host	.	.	.	.	
Celkem	muž		81,703	64,193	77,186	76,684	
Celkem	žena		89,295	74,737	78,130	87,700	
Celkem	profesionál		87,618	81,928	79,167	86,349	
Celkem	host		75,586	63,672	70,997	69,094	
Celkem			85,826	65,211	77,529	81,676	81,676

5.6. Ladění parametrů rozpoznávače

Prvním úkolem po naprogramování rozpoznávače bylo nalezení optimální kombinace všech jeho parametrů. Tento úkol je velmi těžký, protože každý parametr ovlivňuje vliv ostatních parametrů. Bylo nutné pátrat v prostoru o několika rozměrech, tudíž provést několik stovek experimentů. Tolik experimentů však není možné provádět na velkých datech, což zase může vést k tomu, že výsledná sada parametrů bude optimální jen pro data, na kterých byla optimalizována. Rozpoznávač používaný ve všech našich experimentech popsaných v této práci je popsán v kapitole č. 2. Následující parametry ovlivňují kvalitu rozpoznávání:

1. *Akustický model* (skryté markovské modely ve formě buďto 16 nebo 32-mixturových monofónů),
2. *Jazykový model*,
3. *LM Factor* (váha jazykového modelu),
4. *Word Insertion Penalty*,
5. *Prune Threshold*,
6. *Number of Word-End Hypotheses*,
7. *Slovník rozpoznávače*.

Pro ladění parametrů jsme využívali množinu 1 600 vět představenou v kapitole č. 5.5. Po několika úvodních experimentech jsme se rozhodli hledat správnou kombinaci výše vyjmenovaných parametrů v prostoru *Jazykového modelu*, parametru *LM Factor* a *Word Insertion Penalty*. V případě *Akustického modelu* stačí jen malé množství experimentů pro dokázání, že 32-mixturové monofóny dosahují vyšší přesnosti než 16-mixturové monofóny s pouze malým nárůstem spotřebovaného času, viz výsledky v článcích [3] a [4]. Role parametrů *Prune Threshold* a *Number of Word-End Hypotheses* je prořezat strom možných posloupností slov v rozpoznávané větě. Parametr *Prune Threshold* to dělá na úrovni stavů skrytých markovských modelů a parametr *Number of Word-End Hypotheses* to dělá na úrovni slov. O obou parametrech se dá říci, že čím jsou vyšší, tím vyšší je přesnost rozpoznávání, ale každý z nich má jistou úroveň nasycení. Když parametr dosáhne tuto úroveň, přesnost rozpoznávání již stoupá zanedbatelně, ale spotřeba času stoupá stále. *Slovník rozpoznávače* je velmi závislý na testovacích datech. V našich ladících experimentech na množině 1 600 vět jsme používali takzvaný uzavřený slovník. Pojem „uzavřený" znamená, že v něm byla všechna slova z rozpoznávané databáze (v tomto případě pouze slova z ní). Později jsme začali experimentovat s otevřenými slovníky. V těchto experimentech jsme studovali vliv velikosti slovníku, kolokací, dekomponovaných slov a vícenásobných fonetických transkripcí.

Přechod z uzavřeného slovníku 7 033 slov databáze 1 600 vět, se kterou jsme experimentovali v roce 2002 k otevřenému slovníku 312 000 slov, který jsme začali používat v roce 2005, podstatně změnil ideální kombinaci parametrů nalezenou v roce 2002. Náš velký slovník může být testován pouze na velké testovací množině a takový experiment trvá několik hodin. Takže jsme provedli jen několik ladících experimentů, abychom nalezli nové optimální hodnoty parametrů. Tyto hodnoty ukazuje Tabulka č. 3.

Tabulka č. 3 Porovnání optimální množiny parametrů nalezené v roce 2002 pro uzavřený slovník 7 033 slov [10] s hodnotami parametrů používanými v roce 2005 pro otevřený slovník 312 tisíc slov

Parametr	**7 033 slov**	**312 000 slov**
Jazykový model	Witten-Bell	Witten-Bell
LM Factor	6	7
Word Insertion Penalty	– 5	0
Prune Threshold	130	120
Number of Word-End Hypotheses	10	40

Rozdíly v optimálních hodnotách parametrů, které ukazuje Tabulka č. 3, lze vysvětlit podstatným zvětšením velikosti slovníku. V takovém případě klesá průměrná pravděpodobnost bigramů v jazykovém modelu a tak musí být zvětšena váha jazykového modelu (*LM Factor*). Během rozpoznávání musí být také bráno do úvahy větší množství slov, tudíž se musel podstatně zvětšit parametr *Number of Word-End Hypotheses*, možná také parametr *Word Insertion Penalty*.

5.7. Vyhodnocování rozpoznávání

Standardní míry kvality rozpoznávání jsou založeny na rozdílnostech mezi referenčními transkripcemi a výstupem rozpoznávače. Je možné používat následující vzorce ze zdrojů [3], a [2] (str. 271):

$$\textit{Correctness (správnost)}\ [\%] = 100 \cdot \frac{N-D-S}{N} \qquad (6)$$

$$\textit{Accuracy (přesnost)}\ [\%] = 100 \cdot \frac{N-D-S-I}{N} \qquad (7)$$

$$Word\ Error\ Rate\ (chybovost) = WER\ [\%] = 100 \cdot \frac{D+S+I}{N} = 100 - Accuracy \qquad (8)$$

N je celkový počet slov ve správné referenční transkripci.
D (*deletions*) je počet slov, která rozpoznávač vynechal.
S (*substitutions*) je počet slov, která rozpoznávač zaměnil za jiná.
I (*insertions*) je počet slov, která rozpoznávač zapsal navíc.

Hodnoty *D*, *S* a *I* jsou výsledkem algoritmu pro výpočet **minimální editační vzdálenosti** popsané například v knize [2] (str. 153 – 156). Výsledek tohoto algoritmu, suma *D*, *S* a *I* je někdy nazývána **Levenshteinova vzdálenost**. V jedné variantě této míry má každá z těchto tří editačních operací cenu rovnou 1. V jiné variantě má operace vynechání a vložení slova cenu 1 a operace záměny slova má cenu 2, protože je ekvivalentní jednomu vymazání a jednomu vložení slova. Při výpočtu přesnosti (*Accuracy*) rozpoznávání používáme v našich výsledcích tu první variantu.

Jinou důležitou mírou kvality rozpoznávání je spotřeba času. To se vyjadřuje ukazatelem zvaným **real-time factor** *xRT*:

$$xRT = \frac{Čas\ spotřebovaný\ na\ rozpoznávání}{Trvání\ rozpoznávané\ promluvy} \qquad (9)$$

Rychlost rozpoznávání měřená vzorcem (9) je důležitá zvláště pro úlohy, které musí být prováděny v reálném čase, například za účelem opatření živých televizních zpravodajských pořadů titulky. Nevýhodou ukazatele *real-time factor* je jeho závislost na použitém hardwaru. Měl by být vždy doplněn o informaci, na jakém počítači bylo rozpoznávání realizováno.

6. Závěr

6.1. Co bylo v disertační práci vykonáno

Tato práce popsala následující úkoly, které jsou součástí složité úlohy rozpoznávání spojité řeči:

1. Příprava velkého textového korpusu,
2. Sestavení slovníku pro rozpoznávač obsahujícího 312 tisíc slov,
3. Fonetická transkripce slov ve slovníku pro rozpoznávač,
4. Výpočet bigramových jazykových modelů pro rozpoznávač spojité řeči,
5. Příprava 1,5-hodinové testovací databáze promluv,
6. Ladění parametrů rozpoznávače,
7. Vyhodnocování rozpoznávání.

Předmětem těchto úloh byl automatický přepis zpravodajských pořadů v českém jazyce.

6.2. Jaký je přínos této disertační práce pro vědecký obor automatického rozpoznávání řeči

Tato práce popisuje úlohy specifikované v kapitole č. 6.1 podrobněji, než je obvyklé v článcích zabývajících se automatickým rozpoznáváním řeči.

Byl sestaven slovník 800 tisíc nejfrekventovanějších českých slovních tvarů za účelem robustního odhadu pokrytí nezávislého českého textu slovníky o různých velikostech.

Byly popsány tři alternativní metody fonetické transkripce a rozpoznávací experimenty potvrdily důležitost správných fonetických transkripcí ve slovníku rozpoznávače.

Metoda vyhlazování bigramového jazykového modelu zvaná „Witten-Bell discounting“ byla vylepšena a výsledná křížová perplexita takto vyhlazeného jazykového modelu oproti testovacímu korpusu byla skutečně nižší než křížová perplexita počítaná z jazykových modelů vyhlazených alternativními metodami.

Bylo objeveno, že rozložení slov na části ve slovníku a jazykovém modelu rozpoznávače v průměru nevylepšuje přesnost rozpoznávání. Avšak opačný postup – spojování slov, které se často objevují vedle sebe v textu – přesnost rozpoznávání významně vylepšilo.

6.3. Jaký je přínos této disertační práce pro praxi

Výsledky popsané v této práci – textový korpus, slovník, fonetická transkripce, jazykový model a testovací databáze promluv – slouží jako východisko k budoucímu zdokonalování rozpoznávače spojité řeči v laboratoři SpeechLab.

Práce ukázala, že v ní popsané metody řešení mohou pravděpodobně vést ke komerčně využitelným aplikacím pro přepis zpravodajských pořadů v českém jazyce.

6.4. Co by mělo být vykonáno v budoucnu

Zvyšování přesnosti rozpoznávání je výsledkem práce v mnoha oborech. Zde uvádíme pouze úlohy týkající se lingvistické části problému.

1. Mělo by být nalezeno více pravidel pro čištění textového korpusu, zejména pravidla pro přepisování čísel.
2. Tabulka č. 1 v kapitole č. 5.2.3 ukazuje, že přidání jednoho nebo dvou set tisíc nových slov do našeho slovníku pro rozpoznávač, který má zatím 312 tisíc slov, by stále ještě mohlo vylepšit přesnost rozpoznávání.
3. Náš systém pro fonetickou transkripci by měl obsahovat více pravidel a mít je lépe organizována. Měl by být také schopen generovat alternativní fonetické transkripce.
4. Jazykové modely pro náš rozpoznávač byly vždy omezeny kapacitou aktuálně dostupného hardwaru osobních počítačů. Vývoj hardwaru je však tak rychlý, že bychom se měli snažit navrhovat nové jazykové modely implementovatelné na budoucím hardwaru již dnes. Nejslibnějším přístupem podle našeho názoru je vyvíjet jazykové modely, které by se automaticky přizpůsobovaly tématu právě rozpoznávané řeči.
5. Výstup rozpoznávače by měl být co nejbližší psané podobě jazyka. To zahrnuje správné psaní velkých a malých písmen a interpunkce. Také některé číslovky by měly být psány číslicemi a některé slovy. Pro vyřešení tohoto problému musí být sestaveny speciální jazykové modely.

Literatura

[1] Jan Nouza (editor): Počítačové zpracování řeči, cíle, problémy a aplikace. (Sborník článků). Technická univerzita v Liberci. Fakulta mechatroniky a mezioborových inženýrských studií. Katedra elektroniky a zpracování signálů – Laboratoř počítačového zpracování řeči. Liberec 2001, ISBN 80-7083-551-6.

[2] Daniel Jurafsky, James H. Martin: Speech and Language Processing. Prentice Hall, Inc., New Jersey, 2000, ISBN 0-13-095069-6.

[3] Dana Nejedlová, Jan Nouza: Language Model Support for Continuous Speech Recognition in Czech Language. In: Signal Processing, Pattern Recognition, and Application, Anaheim (USA), Calgary (Kanada), Curych (Švýcarsko) 2002, ISBN 0-88986-338-5, str. 541 – 546, ISSN 1482-7921.

[4] Jan Nouza: Strategies for Developing a Real-Time Continuous Speech Recognition System for Czech Language. In: Text, Speech and Dialogue (eds. Petr Sojka, Ivan Kopeček, Karel Pala) Springer-Verlag, Heidelberg, 2002, str. 189 – 196, ISBN 3-540-44129-8, ISSN 0302-9743.

[5] Dana Nejedlová, Jindra Drábková, Jan Kolorenč, Jan Nouza: Lexical, Phonetic, and Grammatical Aspects of Very-Large-Vocabulary Continuous Speech Recognition of Czech Language. In: Electronic Speech Signal Processing, Proceedings of the 16th Conference on Electronic Speech Signal Processing joint with the 15th Czech-German Workshop on Speech Processing, Dresden, Německo, září 2005, TUDpress, str. 224 – 231, ISBN 3-938863-17-X, ISSN 0940-6832.

[6] Pavel Ircing: Large Vocabulary Continuous Speech Recognition of Highly Inflectional Language (Czech). [Disertační práce] Západočeská univerzita v Plzni. Fakulta aplikovaných věd. Plzeň 2003.

[7] Jan Nouza, Jindřich Žďánský, Petr David, Petr Červa, Jan Kolorenč, Dana Nejedlová: Fully Automated System for Czech Spoken Broadcast Transcription with Very Large (300K+) Lexicon. In: Interspeech, Lisabon-Portugalsko, 2005, ISCA, Bonn, Německo, str. 1681 – 1684, ISSN 1018-4074.

[8] Dana Nejedlová: Building and Evaluation of a Large Vocabulary for a Czech Voice Dictation System. In: ECMS (The 6th International Workshop on Electronics, Control, Measurement and Signals), Liberec, 2003, str. 74 – 78, ISBN 80-7083-708-X.

[9] Josef Psutka: Komunikace s počítačem mluvenou řečí, Academia, Prague 1995, ISBN 80-200-0203-0.

[10] Dana Nejedlová: Comparative Study on Bigram Language Models for Spoken Czech Recognition. In: Text, Speech and Dialogue (eds. Petr Sojka, Ivan Kopeček, Karel Pala) Springer-Verlag, Heidelberg, 2002, str. 197 – 204, ISBN 3-540-44129-8, ISSN 0302-9743.

[11] Jan Nouza, Dana Nejedlová, Jindřich Žďánský, Jan Kolorenč: Very Large Vocabulary Speech Recognition System for Automatic Transcription of Czech Broadcast Programs. In: ICSLP (eds. Soon Hyob Kim and Dae Hee Youn), Sunjin Printing Co., 2004, str. 409 – 412, ISSN 1225-441x.

Vlastní publikované práce

Dana Nejedlová, Jindra Drábková, Jan Kolorenč, Jan Nouza: "Lexical, Phonetic, and Grammatical Aspects of Very-Large-Vocabulary Continuous Speech Recognition of Czech Language". Prezentováno na 16. konferenci „Electronic Speech Signal Processing“ spojené s „15th Czech-German Workshop on Speech Processing“ Ústavu radiotechniky a elektroniky Akademie věd České republiky v Lichtenštejnském paláci v Praze 27. září 2005. In: Electronic Speech Signal Processing, Proceedings of the 16th Conference on Electronic Speech Signal Processing joint with the 15th Czech-German Workshop on Speech Processing, Drážďany, Německo, Září 2005, TUDpress, str. 224 – 231, ISBN 3-938863-17-X, ISSN 0940-6832.

Jan Nouza, Jindřich Žďánský, Petr David, Petr Červa, Jan Kolorenč, Dana Nejedlová: "Fully Automated System for Czech Spoken Broadcast Transcription with Very Large (300K+) Lexicon". In: proc. 9th European Conference on Speech Communication and Technology Interspeech 2005 (CD-ROM), Lisabon, Portugalsko, 2005, ISCA, Bonn, Německo, str. 1681 – 1684, ISSN 1018-4074.

Jan Nouza, Dana Nejedlová, Jindřich Žďánský, Jan Kolorenč: "Very Large Vocabulary Speech Recognition System for Automatic Transcription of Czech Broadcast Programs". In: proc. 8th International Conference on Spoken Language Processing ICSLP 2004 (editoři: Soon Hyob Kim and Dae Hee Youn) (4 svazky a CD-ROM), Jeju Island, Korea, říjen 2004, Sunjin Printing Co., str. 409 – 412, ISSN 1225-441x.

Dana Nejedlová: "Lexicon and Language Model Building for Czech Very-Large-Vocabulary Speech Recognition". Prezentováno na „The 14th Czech-German Workshop on Speech Processing“ Ústavu radiotechniky a elektroniky Akademie věd České republiky v Karlově univerzitě v Praze 14. září 2004. In: Speech Processing, 14th Czech-German Workshop, Praha 2004, str. 82 – 92, ISBN 80-86269-11-6.

Dana Nejedlová: "Construction of a Dictation System for Czech Physicians". Prezentováno na „The 13th Czech-German Workshop on Speech Processing“ Ústavu radiotechniky a elektroniky Akademie věd České republiky v Karlově univerzitě v Praze 16. září 2003. In: Speech Processing, 13th Czech-German Workshop, Praha 2004, str. 115 – 117, ISBN 80-86269-10-8.

Dana Nejedlová, Jan Nouza: "Building of a Vocabulary for the Automatic Voice-Dictation System". Prezentováno na 6. mezinárodní konferenci TSD 2003 v Českých Budějovicích 9. září 2003. In: Text, Speech and Dialogue (editoři: Václav Matoušek, Pavel Mautner) Springer-Verlag, Heidelberg, 2003, str. 301 – 308, ISBN 3-540-20024-X, ISSN 0302-9743.

Dana Nejedlová: "Building and Evaluation of a Large Vocabulary for a Czech Voice Dictation System". Prezentováno na „The 6th International Workshop on Electronics, Control, Measurement and Signals – ECMS 2003“ 3. června 2003. In: ECMS 2003, Liberec, June 2003, str. 74 – 78, ISBN 80-7083-708-X.

Dana Nejedlová: "Building a 20K Vocabulary and Language Model for Czech Language". In: Speech Processing, 12th Czech-German Workshop, Praha 2002, str. 67 – 70, ISBN 80-86269-09-4.

Dana Nejedlová: “Comparative Study on Bigram Language Models for Spoken Czech Recognition”. Prezentováno na 5. mezinárodní konferenci TSD 2002 v Brně 9. září 2002. In: Text, Speech and Dialogue (editoři: Petr Sojka, Ivan Kopeček, Karel Pala) Springer-Verlag, Heidelberg, 2002, str. 197 – 204, ISBN 3-540-44129-8, ISSN 0302-9743.

Dana Nejedlová, Jan Nouza: “Language Model Support for Continuous Speech Recognition in Czech Language”. Prezentováno na „The IASTED International Conference SPPRA 2002” v Řecku na ostrově Kréta 27. června 2002. In: Signal Processing, Pattern Recognition, and Application, Anaheim (USA), Calgary (Kanada), Curych (Švýcarsko) 2002, str. 541 – 546, ISBN 0-88986-338-5, ISSN 1482-7921.

Jan Nouza, Dana Nejedlová: “Experiments with Read Speech Recognition in Czech”. Prezentováno na „The 11th Czech-German Workshop on Speech Processing“ Ústavu radiotechniky a elektroniky Akademie věd České republiky v Karlově univerzitě v Praze 18. září 2001. In: Speech Processing, 11th Czech-German Workshop, Praha 2001, str. 46 – 49, ISBN 80-86269-07-8.

Dana Nejedlová, Marek Volejník: „Transkripce psaného českého textu do fonetické podoby“. In: Počítačové zpracování řeči – cíle, problémy, metody a aplikace (symposium), Technická univerzita v Liberci, Liberec 2001, str. 10 – 22, ISBN 80-7083-551-6.

Dana Nejedlová, Jan Nouza: “Phonetic Transcription of Czech Language Using a NETtalk-type Neural Network”. Prezentováno na „The 10th Czech-German Workshop on Speech Processing“ Ústavu radiotechniky a elektroniky Akademie věd České republiky v Karlově univerzitě v Praze 20. září 2000. In: Speech Processing, 10th Czech-German Workshop, Prague 2000, str. 37 – 40, ISBN 80-86269-05-1.