Speech Technologies for Serbian and Kindred South Slavic Languages

Vlado Delić[1], Milan Sečujski[1], Nikša Jakovljević[1], Marko Janev[1,2], Radovan Obradović[1,2] and Darko Pekar[2]
[1]*Faculty of Technical Sciences, University of Novi Sad,*
[2]*AlfaNum – Speech Technologies, Novi Sad,*
Serbia

1. Introduction

This chapter will present the results of the research and development of speech technologies for Serbian and other kindred South Slavic languages used in five countries of the Western Balkans, carried out by the University of Novi Sad, Serbia in cooperation with the company AlfaNum. The first section will describe particularities of highly inflected languages (such as Serbian and other languages dealt with in this chapter) from the point of view of speech technologies. The following sections will describe the existing speech and language resources for these languages, the automatic speech recognition (ASR) and text-to-speech synthesis (TTS) systems developed on the basis of these resources as well as auxiliary software components designed in order to aid this development. It will be explained how the resources originally built for the Serbian language facilitated the development of speech technologies in Croatian, Bosnian, and Macedonian as well. The paper is concluded by the directions of further research aimed at development of multimodal dialogue systems in South Slavic languages.

1.1 Particularities of highly inflected languages

The complexity of a number of tasks related to natural language processing is directly related to the complexity of the morphology of the language. The principal feature of inflective languages is that words are modified in order to express a wide range of grammatical categories such as tense, person, number, gender and case. Together with a high degree of derivation with the use of prefixes and suffixes typical for such languages, this results in extremely large vocabularies. As a consequence, statistically oriented language models (based on *N*-grams), which are quite successful in modelling languages with a modest degree of morphological complexity, turn out to be inadequate for use for morphologically more complex languages without significant modifications (Jurafsky & Martin, 2000).

The problem affects both automatic speech recognition and text-to-speech synthesis. In the case of ASR, extremely large vocabularies require the existence of extremely large corpora for obtaining robust *N*-gram statistics. For instance, a corpus of English containing 250.000 tokens actually contains approximately 19.000 types (Oravecz & Dienes, 2002), while a

Source: Advances in Speech Recognition, Book edited by: Noam R. Shabtai,
ISBN 978-953-307-097-1, pp. 164, September 2010, Sciyo, Croatia, downloaded from SCIYO.COM

corpus of Serbian of the same size contains approximately 46.000 types (Sečujski, 2009). Furthermore, the rate of out-of-vocabulary (OOV) words is also much higher in case of morphologically rich languages. A number of solutions to this problem have been proposed, mostly based on modelling the statistics of subword units instead of words. Some of the proposed solutions even target South Slavic languages (Sepesy Maučec et al., 2003), however, none of them results in a system of an accuracy sufficient for its practical usability.

The impact of the problem with respect to TTS is related to the difficulty of accurate high-level synthesis. For the text to be delivered to the listener as intelligible and natural-sounding speech, it has to be pre-processed, and most of the activities included require some kind of estimation of robust statistics of the language, as it will be explained in more detail in the following sections. As was the case with ASR, the size of the vocabulary leads to data sparsity, resulting in the need for significantly greater corpora sufficient for obtaining a language model of the same robustness in comparison to languages with a simpler system of morphological categories.

When the four South Slavic languages used in the Western Balkans (namely: Serbian, Croatian, Bosnian, Macedonian) are examined, it can be seen that they exhibit extreme similarities at levels ranging from phonetic and morphological to syntactic and semantic. With the exception of Macedonian, all these languages have until recently been considered as variants of a single language (Serbo-Croatian). Owing to this fact, tools and procedures used for development of most of the resources originally developed for Serbian (including a morphological dictionary (Sečujski, 2002), a morphologically annotated corpus (Sečujski, 2009) and an expert system for part-of-speech tagging (Sečujski, 2005)) were re-used to develop corresponding resources for the other languages. In some cases it was possible to easily create the resources for the other languages by simple modification of existing resources for Serbian, as will be explained in more detail in the following sections.

2. Text-to-Speech

This section will describe AlfaNum TTS, the first fully functional text-to-speech synthesiser in Serbian language, which has been adapted to Croatian, Bosnian and Macedonian as well. It is constantly being improved by introducing novel techniques both at high and low synthesis level (Sečujski et al., 2007).

The high-level synthesis module includes processing of text and its conversion into a suitable data structure describing speech signal to be produced. The output of the high-level synthesis module is a narrow phonetic transcription of the text, containing the information on the string of phonemes/allophones to be produced as well as all relevant prosody information, such as f_0 movement, energy contour and temporal duration of each phonetic segment. The principal modules of a high-level synthesis module are given in Fig. 1.

2.1 High-level synthesis

The text preprocessing module is charged with conversion of text into a format more suitable for text analysis. The text to be preprocessed is usually in a plain format, not even tagged for ends of sentences, and it is up to the *sentence boundary detection module* to locate sentence boundaries, which is the first stage of preprocessing. Most practical systems use heuristic sentence division algorithms for this purpose, and although they can work very well provided enough effort was put in their development, they still suffer from the same

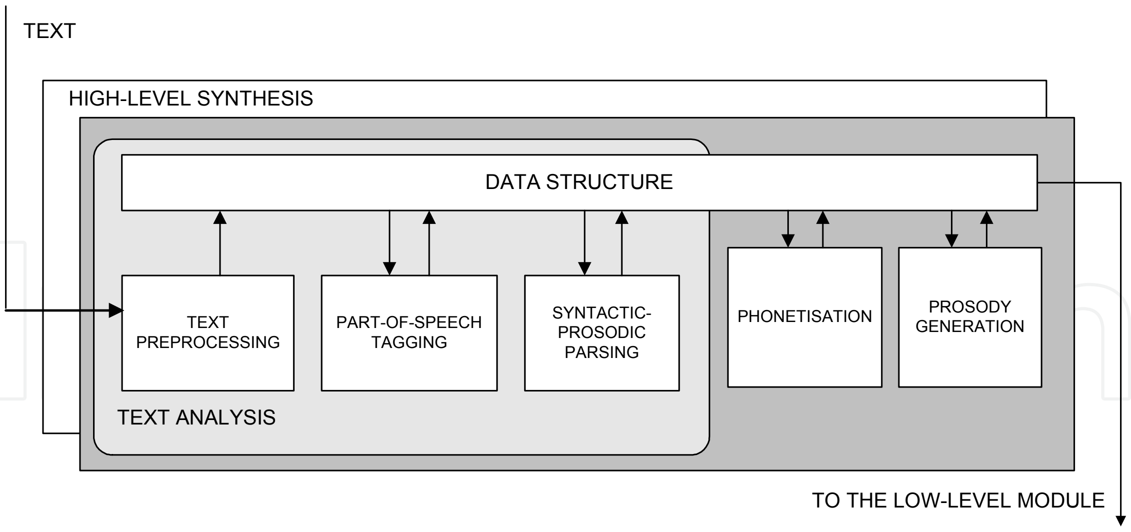


Fig. 1. An overview of the high-level speech synthesis module.

problems of heuristic processes in general – they require a lot of hand-coding and domain knowledge on the part of the person developing the module. Besides neural networks and maximum entropy models, the framework of statistical classification trees can also be effectively used for this purpose, as was first shown in (Riley, 1989). Furthermore, it can be made more powerful by introduction of specialised linguistically motivated features in tree construction. Although the sentence boundary detection module currently used within the AlfaNum TTS system (Sečujski et al., 2002) is a purely heuristic one, development of a tree-based classifier for sentence boundary detection is under way. Further preprocessing stages include conversion of a long string of characters (including whitespaces) into lists of words. Texts, however, do not consist of orthographic words only, and all non-orthographic expressions have to be expanded into words. The preprocessor is thus also charged with processing of punctuation marks, handling acronyms and abbreviations and transcribing numbers into literals. Each of these problems represents a highly language-dependent research area. All of the preprocessing modules currently used by the AlfaNum TTS system for these purposes are of heuristic nature.

Another source of problems is that the surface form of a word is not always a sufficient source of information as to how the word should be read. There is a number of morphological and syntactical ambiguities to be resolved for the word to be read correctly. The critical properties of each word from the point of its conversion into speech are its phonetic transcription as well as the position of accent(s) within it. In the case of all of the aforementioned languages the task of phonetisation is (nearly) trivial, as in each of them one letter basically corresponds to one sounds. The phonology of these languages is rather complex as there are numerous interactions between phonemes at morpheme boundaries, however, almost all of these interactions are reflected in writing as well, and thus do not represent a problem as regards TTS. On the other hand, from the point of view of stress position and type, the situation is less favourable. For example, Serbian, Croatian and Bosnian have an extended system of accentuation, which, from the phonological point of view, has four accents divided into two groups according to their quantity and quality: *long-fall*, *short-fall*, *long-rise* and *short-rise*, their exact realisation varying according to vernacular. Assigning an erroneous accent to a word would affect speech perception to the point that sometimes a completely different meaning would be perceived from the utterance. The accentuation of Macedonian is somewhat simpler. Besides recent loanwords, word stress in Macedonian is antepenultimate, which means that it falls on the third from last syllable in words with three or more syllables, and on the first syllable in other words. Thus, in most cases, reasonably correct pronunciation of a word does not require its full morpho-syntactic disambiguation.

In general, most of the morpho-syntactic disambiguation required for correct rendering of a word is done through part-of-speech (POS) tagging (although in the case of all of the aforementioned languages there is an occasional dependence of accent type or position on syntax as well). Within the POS tagging procedure, each word has to be assigned some specific additional information related to its morphological status, contained in a unique morphological descriptor or part-of-speech (POS) tag. In case of languages with complex morphology, such tags usually have specified internal structure, and their total number (tagset size) is much larger than in case of languages with simpler morphology (Hajič & Hladká, 1998). This, in turn, leads to the well-known problem of data sparsity, i.e. the fact that the amount of training data necessary increases rapidly with tagset size, making highly accurate part-of-speech taggers for such languages extremely hard to obtain. Whichever of the statistical tagging techniques is used, a number of modifications become necessary when dealing with highly inflective or agglutinative languages (Jurafsky & Martin, 2000). The AlfaNum TTS system performs POS tagging by using a technique that is based on performing a beam-search through a number of partial hypotheses, evaluating them with respect to a database of linguistic rules (Sečujski, 2005). The basic set of rules were hand-coded, however, the database has since been significantly augmented using a transformational-based tagger.

For any partial hypothesis to be considered, the system must know the possible tags for each surface form. However, they cannot be deduced from the surface form itself, which points to the conclusion that any strategy aiming at accurate POS tagging and accent assignment should rely on morphologically oriented dictionaries.

Within this research, by using a software tool created for that purpose, the AlfaNum morphological dictionary of Serbian language was created, containing approximately 100.000 lexemes at this moment, i.e. approximately 3.9 million inflected forms. The research described in this chapter also required that an extensive part-of-speech tagged text corpus be built. Within this research, by using another software tool created for that purpose, the AlfaNum Text Corpus (ATC) was created and part-of-speech tagged, containing approximately 11.000 sentences with approximately 200.000 words in total. Based on the same principles, a Croatian dictionary of approximately the same size was subsequently developed. Owing to extreme similarities of Serbian, Bosnian and Croatian, the Serbian and Croatian dictionaries are jointly used for tagging of Bosnian, and instead of full tagging of Macedonian, only stress assignment is carried out, according to the rule of the antepenultimate syllable and a dictionary of exceptions containing approximately 44.000 types.

Each entry in the AlfaNum morphological dictionary of Serbian, besides the morphological descriptor, also contains the data related to the accentuation of the word, as well as the lemma (base form), which is useful for lemmatisation. The term *entry* thus denotes a particular inflected form of a word, together with the corresponding lemma, values of part-of-speech and morphological categories, as well as its accent structure (a string of characters denoting accent type associated to each syllable). An example of an entry would be:

Vb-p-1-- *uzećemo* (*uzeti*) [\-00].

Morphological categories that are marked are dependent on the part-of-speech, and thus e.g. verbs are marked for tense/mood, gender, number and person, but only in case a particular category is applicable to the tense/mood in question. The example above represents a verb (V) in 1st person (1) plural (p) of the future tense (b), whose surface form is *uzećemo* and whose base form is *uzeti*. The data related to accentuation are given in square brackets. In this way, all the inflected forms of words are present in the dictionary, and the task of part-of-speech tagging of an unknown text amounts (in most cases) to the selection of the correct tag out of all possible tags provided by the dictionary, rather than actual morphological analysis of words.

The dictionary was built in an efficient way using a software tool previously developed for that purpose (Sečujski, 2002). This tool is based on direct implementation of inflectional paradigms of the Serbian language, and its application enables efficient input of complete paradigms instead of individual entries.

When all the possible tags are provided by the dictionary, it remains to select the correct one. As it would be impossible to consider all tag combinations separately, an algorithm similar to dynamic programming is used, keeping the number of partial hypotheses under control.

Let us consider a sentence $W = w_1w_2...w_N$. Each of the words w_i has a corresponding tag list:

$$T_i = \{t_{i1}, t_{i2}, ... t_{iN_i}\}, \tag{1}$$

and its actual tag t_i is one of the t_{ij}, $j = 1, 2,... N_i$. Initially only the hypotheses of length one are considered, containing only the first word of the sentence:

$$H_1 = \{(t_{11}), (t_{12}), ... (t_{1N_1})\}. \tag{2}$$

In every following step of the algorithm, each variant of the next word is combined with each of the existing partial hypotheses. A set of all possible hypotheses of length two is thus:

$$H_2 = \{(t_{1m}, t_{2n}) \mid m = 1, 2... N_1, n = 1, 2... N_2\}. \tag{3}$$

Each time a new word is appended in such a way, the score of each hypothesis is recalculated, based on the likelihood that a word with such a tag can follow. If the number of all hypotheses exceeds a previously set limit L, only L hypotheses with highest scores are retained, and all the others are discarded. The procedure continues until all words are included and the hypothesis with the highest score is selected as the estimate of actual tag sequence $T = t_1t_2...t_N$. Fig. 2 shows an example of such analysis. The algorithm described here performs in time proportional to the length of the sentence, and one of its interesting features is that it produces partial results very quickly. The first word in the sentence is assigned its tag long before the analysis is over, which is consistent with the notion that, when reading a sentence, humans are usually able to start pronouncing it far before they reach its end, and that they organise the sentence into simple prosodic units which can be obtained from local analysis (Dutoit, 1999). Furthermore, this feature of the algorithm is especially useful from the point of view of speech synthesis, because synthesis of the speech signal can start as soon as the first partial results are obtained, which minimises the delay introduced by POS tagging.

The initial criteria for actual scoring of the hypotheses are based on rules defined according to the statistics of different parts-of-speech in Serbian language and grammatical rules found

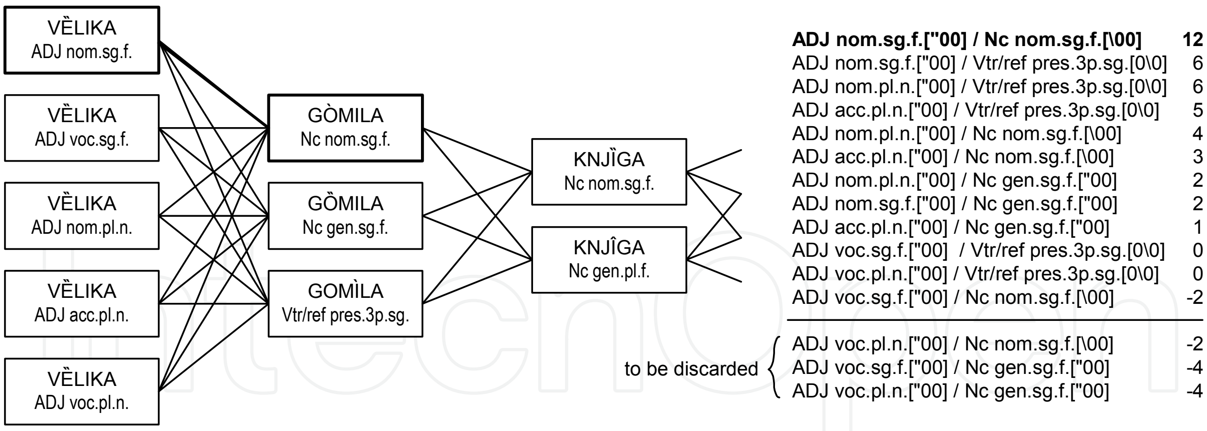

VȄLIKA
ADJ nom.sg.f.
VȄLIKA
ADJ voc.sg.f.
VȄLIKA
ADJ nom.pl.n.
VȄLIKA
ADJ acc.pl.n.
VȄLIKA
ADJ voc.pl.n.
GÒMILA
Nc nom.sg.f.
GȌMILA
Nc gen.sg.f.
GOMÌLA
Vtr/ref pres.3p.sg.
KNJÌGA
Nc nom.sg.f.
KNJÎGA
Nc gen.pl.f.
ADJ nom.sg.f.["00] / Nc nom.sg.f.[\00] 12
ADJ nom.sg.f.["00] / Vtr/ref pres.3p.sg.[0\0] 6
ADJ nom.pl.n.["00] / Vtr/ref pres.3p.sg.[0\0] 6
ADJ acc.pl.n.["00] / Vtr/ref pres.3p.sg.[0\0] 5
ADJ nom.pl.n.["00] / Nc nom.sg.f.[\00] 4
ADJ acc.pl.n.["00] / Nc nom.sg.f.[\00] 3
ADJ nom.pl.n.["00] / Nc gen.sg.f.["00] 2
ADJ nom.sg.f.["00] / Nc gen.sg.f.["00] 2
ADJ acc.pl.n.["00] / Nc gen.sg.f.["00] 1
ADJ voc.sg.f.["00] / Vtr/ref pres.3p.sg.[0\0] 0
ADJ voc.pl.n.["00] / Vtr/ref pres.3p.sg.[0\0] 0
ADJ voc.sg.f.["00] / Nc nom.sg.f.[\00] -2
to be discarded
ADJ voc.pl.n.["00] / Nc nom.sg.f.[\00] -2
ADJ voc.sg.f.["00] / Nc gen.sg.f.["00] -4
ADJ voc.pl.n.["00] / Nc gen.sg.f.["00] -4

Fig. 2. An example of a step in the disambiguation algorithm for the sentence "*Velika gomila knjiga stoji na stolu*". The diagram shows the situation after all the hypotheses of length two are considered, and three of them with lowest scores are to be discarded (in this example stack size limit is L = 12).

in the literature. Further error-correcting rules have been discovered using the transformational-based part-of-speech tagger described in (Sečujski, 2009), and trained on individual sections of the AlfaNum Text Corpus. The tagger is based on the general transformation-based learning paradigm (Brill, 1992), but enhanced with certain learning strategies particularly applicable to highly inflected languages (Sečujski, 2009). Both hand-coded and automatically obtained rules are created following standard templates such as:

Award n points to a partial hypothesis $h = (w_1, w_2,... w_l)$:

- If w_l is tagged t_i
- If w_l is tagged t_i and w_{l-1} is tagged t_j
- If w_k is tagged t_i, w_{l-1} is tagged t_j and w_{l-2} is tagged t_k
- If w_l is tagged t_i and w_{l-1} is tagged t_j and the value of a morphologic category c contained in the tag t_i is the same (is not the same) as the value of the corresponding morphologic category contained in the tag t_j
- If w_l is tagged t_i and w_{l-1} is tagged t_j and all of the values of morphologic categories c_1, c_2,...c_k contained in the tag t_i are the same (are not the same) as the values of corresponding morphologic categories contained in the tag t_j

where n is assigned depending on the technique used.

After the (presumably) correct tag sequence has been discovered, the next step consists of modifying accent patterns to account for occasional dependence of accent type and/or position on syntax, as described previously, and performing syntactic-prosodic parsing of the sentence (detecting prosodic events such as major and minor phrase breaks, setting sentence focus etc.). Both are currently done using heuristic algorithms, however, the development of a tree-based classifier which would be in charge of the latter is under way. This classifier will be trained on sections of the AlfaNum Text Corpus which are annotated for minor and major phrase breaks as well as sentence focus.

It remains to assign each word its actual prosodic features, such as durations of each phonetic segment as well as f_0 and energy contours. In the version for the Serbian language, this is currently performed using regression trees trained on the same speech database used for speech synthesis. The section of the database used for training of regression trees is fully annotated with phone and word boundaries, positions of particular accent types and prosodic events such as major and minor phrase breaks and sentence focus. Separate regression trees are used for prediction of phonetic durations and for prediction of f_0 and energy contours. Owing to this approach, actual acoustic realisation of each accent in synthesised speech is expected to correspond to the most common realisation of the same accent in a phonetically and prosodically similar context in the speech database. The listening experiments carried out so far have confirmed the expectation that such an approach would lead to superior naturalness of synthetic speech in comparison with the previous version, which was based on heuristic assignment of predefined f_0 and energy contours corresponding to particular accentuation configurations (Sečujski et al., 2002). The versions of the synthesiser for Croatian, Bosnian and Macedonian language still use the heuristic algorithm for prosody prediction, however, the Croatian synthesiser is expected to switch to regression-tree based prosody prediction soon, as prosodic annotation of the Croatian speech database is currently under way. As was the case with morphological dictionaries, significant experience in creation of other resources for the Serbian language will certainly contribute to efficient creation of appropriate resources for other kindred languages as well.

2.2 Low-level synthesis

The term low-level synthesis refers to the actual process of producing a sound that is supposed to imitate human speech as closely as possible, based on the output of the high-level synthesis module described in the previous subsection. In all of the available versions of the system, the concatenative approach has been used as being the simplest and at the same time offering high intelligibility and reasonably high flexibility in modifying prosodic features of available phonetic segments prior to synthesis (Sečujski et al., 2002).

The AlfaNum R&D team has recently recorded a new speech database containing 10 hours of speech from a single speaker (instead of a 2.5 hour database previously used), and so far annotated approximately 3 hours of it using visual software tools specially designed for that purpose (Obradović & Pekar, 2000). By keeping score of the identity of each phone in the database and its relevant characteristics (such as the quality of articulation, nasalisation and vocal fry), use of phones in less than appropriate contexts was discouraged, which further contributed to overall synthesised speech quality. Unlike most other synthesisers developed for kindred languages so far, the AlfaNum TTS engine can use larger speech segments from the database, according to both phonetic and prosodic requirements, and select them at runtime in order to produce the most intelligible and natural-sounding utterance for a given plain text (Beutnagel et al., 1999). The full increase in synthesis quality is yet to come after the remaining 7 hours of speech are annotated.

According to differences between the existing and the required values of parameters previously defined, each speech segment which can be extracted and used for synthesis is assigned *target cost*, and according to differences at the boundaries between two segments, each pair of segments which can be concatenated is assigned *concatenation cost*. Target cost is the measure of dissimilarity between existing and required prosodic features of segments, including duration, f_0, energy and spectral mismatch. Concatenation cost is the measure of mismatch of the same features across unit boundaries. The degree of impairment of phones is also taken into account when selecting segments, as explained previously. The task of the synthesiser is to find a best path through a trellis which represents the sentence, that is, the path along which the least overall cost is accumulated. The chosen path determines which segments are to be used for concatenation, as shown in Fig. 3, with s_{ij} denoting segments, c'_{ij}

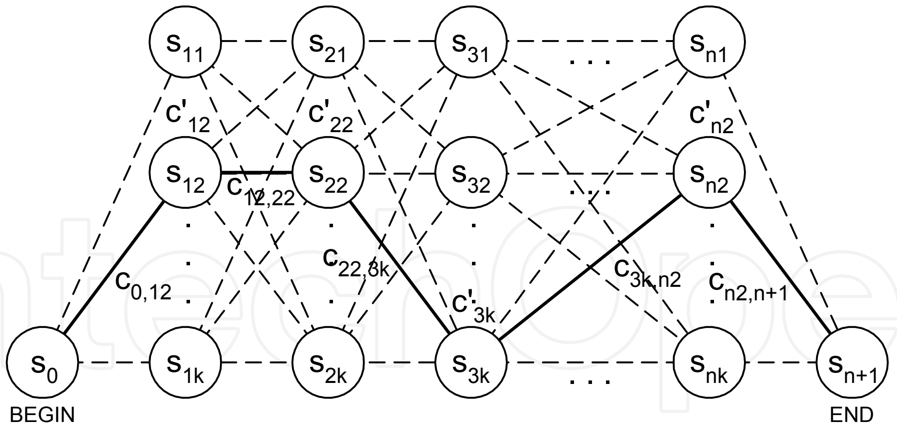


Fig. 3. Finding the best path through a trellis representing a sentence.

denoting segment costs and $c_{ij,pq}$ denoting concatenation costs. Segment modifications related to smoothing and prosody manipulation are carried out using the TD-PSOLA algorithm.

In a version which is currently under development, an alternative to the TD-PSOLA low-level synthesis algorithm is being introduced – HMM based synthesis (Tokuda et al., 2000). Segmental intelligibility tests have still to be carried out, yet the first results seem to be encouraging.

3. Automatic speech recognition

AlfaNum automatic speech recognition (ASR) system as well as most of state-of-the-art systems is based on hidden Markov models (HMM). State emitting probabilities are modelled by Gaussian mixture models (GMM), with each Gaussian distribution defined by its mean and full covariance matrix. The parameters of each Gaussian in GMM are estimated using the Quadratic Bayesian classifier (Webb, 1999), which is a generalisation of the standard K-means classification iterative procedure. The goal of decoding in the AlfaNum ASR systems is to find the most probable word sequence corresponding to the input speech, as well as a confidence measure for each recognition. Viterbi algorithm is used for a search for the most probable word sequence. To accelerate the search procedure, beam search and Gaussian selection (Janev et al., 2008) are used.

3.1 Speech corpus

One of the first steps in development of an ASR system is speech corpus acquisition. Since 1998 a speech corpus has been developed for Serbian according to the SpeechDat(E) standard (Delić, 2000). It contains utterances from about 800 native speakers (400 male and 400 female), which have been recorded via the public switched telephone network. Today, the corpus volume is about 12 hours of speech (silent and damaged segments are excluded). A section of the corpus, containing 30 minutes of speech from about 180 speakers (100 male and 80 female), is used as the test set for the experiments. Transcriptions are at the phone level, and boundaries between phones are corrected manually (Obradović & Pekar, 2000). The language of the speech corpus is Serbian, but it is used for development of ASR applications in Croatian and Bosnian as well, since the phonetic inventories of these kindred languages are practically identical, with minor variations in pronunciation of certain phonemes.

3.2 Acoustic models

For the purposes of ASR, several changes had to be introduced into the phonetic inventory of the Serbian language. Instead of the standard 5 vowels in Serbian i.e. /i/, /e/, /a/, /o/ and /u/ (IPA notation), two sets containing 5 long and 5 short vowels are taken into consideration. This distinction has been motivated by the fact that short vowels usually do not reach its target position. A vowel is marked as long, if its duration is longer than 75 ms and its average energy is greater than 94% of average vowel energy in the utterance containing the vowel, otherwise the vowel is marked as short. Phone /ə/ is regarded as a standard vowel as well. Moreover, closure and explosion (friction) of stops (affricates) are modelled separately in order to obtain more precise initial models. These models will be referred as sub-phones in further text.

Acoustic features of phone are influenced by articulatory properties of nearby phones, and this influence is called coarticulation. In order to capture acoustic variations of phone caused by coarticulation, triphone (context dependent phone/sub-phone) is used as basic modelling unit (Young et. al., 1994). Introducing sub-phone models results in the slightly complex procedure for conversion of words into appropriate sequence of triphones, where sub-phone models are treated as a single phone. Silence and non-speech sounds (various types of impulse noise) are modelled as context independent units.

The number of HMM states per model is proportional to the average duration of all the instances of the corresponding phone in the training database (e.g. long vowels are modelled by five states and stop explosions by only one state). On this way slightly better modelling of path in feature space is achieved at the cost of reducing the number of observations per state.

The number of mixtures per HMM state is determined semi-automatically. It gradually increases until the average log likelihood on the validation set starts to decrease or the maximum number of mixtures for the given state is reached. Maximum number of mixtures per state depends on which model that state belongs. For example, models for fricatives /s/ and /ʃ/ have fewer mixtures per state than vowels, because the coarticulation effects on these fricatives are smaller than on vowels.

Using triphones instead of monophones leads to a very large set of models and insufficient training data for each triphone. All HMM state distributions would be robustly estimated if sufficient observations were available for each state. This could be achieved by extending the training corpus or by including observations related to acoustically similar states. The second solution, known as tying procedure, was chosen as being less expensive, even though it generates some suboptimal models.

3.3 Tying procedure

The main issue in the tying procedure is how to define acoustically similar states. The vocal articulators are moved at relatively slow speeds and do not remain in the steady positions through the duration of a phone. They are moving from the position required to articulate the preceding phone to the position required for the successive phone, via the position needed for the current phone. Therefore, acoustically similar states are the states of the same phone at the same position in HMM (left-to-right model topology is used), which have phones with a similar place and manner of articulation in their context. The level of the state similarity depends on the similarity of its contexts. The previous phone has more influence on the initial HMM states than on the final HMM states, and subsequent phone has more influence on the final states than on the initial. Hence the position of the state in HMM defines the importance of the context. For the initial state, and all states close to it, the left context is more important, and for the final state, all states close to it and central state (if such a state exists) the right context is more important (See example in Fig. 4.). It is obvious that the states with the same more important context and a different less important context are more similar than vice versa (Young et. al., 1994).

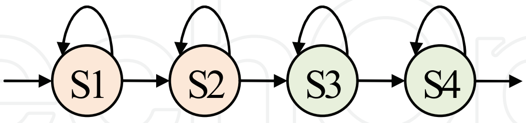


Fig. 4. Left-to-right HMM topology with 4 emitting states. Important context for states 1 and 2 is left, and for states 3 and 4 is right context.

For the tying procedure, it is necessary to define phone similarity. Definition of phone similarity is based on our linguistic knowledge about the place and manner of articulation of the phone. Fig. 5. illustrates similarity level tree. IPA notation is used for the phone labels. Non-speech sounds like silence, background noise and unarticulated sounds are marked by 'sil', 'int' and 'unk', respectively.

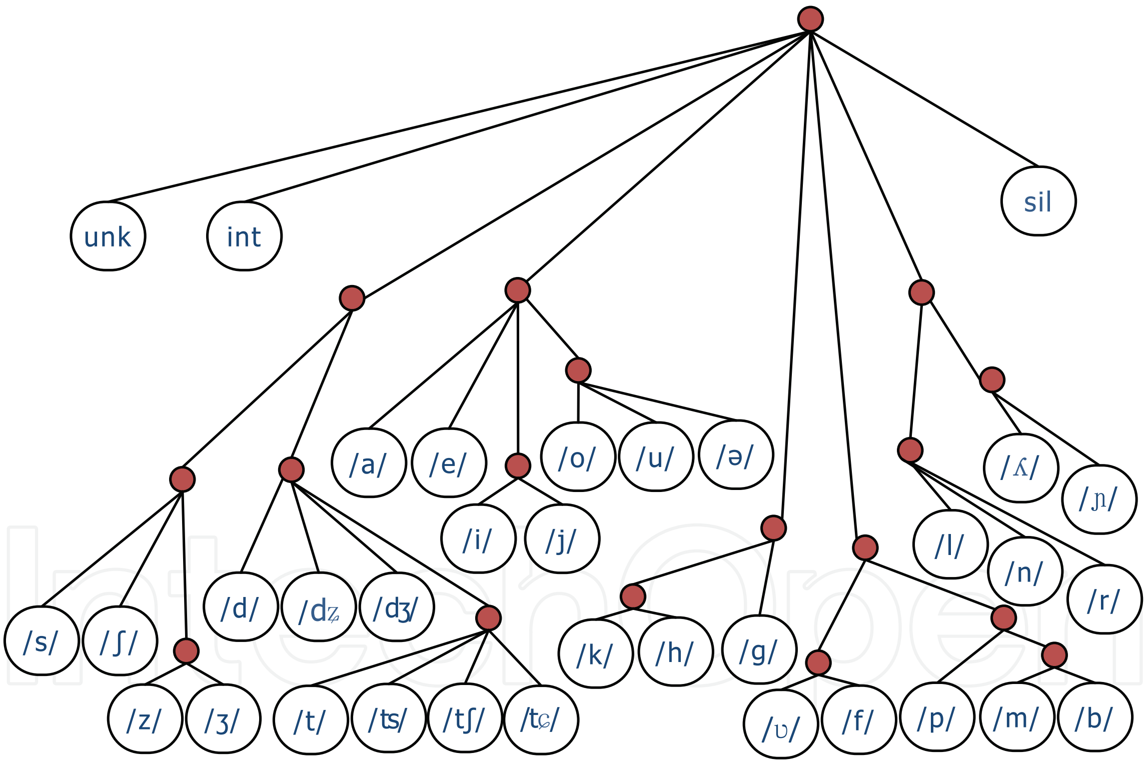


Fig. 5. The tree of the phonetic similarity. Closure and explosion (friction) of stops (affricates) are treated as single context.

The tying procedure (Fig. 6) is applied only to the states with an insufficient number of observations. Mark with *Si* the *i*-th HMM state of the phone *Ph* (*i* is the indicator of state position in left-to-right HMM topology as well). The more important context for the state *Si* is *MIC* and less important context is *LIC*. Suppose that *Si* has an insufficient number of observations for robust parameter estimation. The proposed algorithm attempts to obtain the additional observations for the state *Si*, by borrowing observations from the *i*-th states,

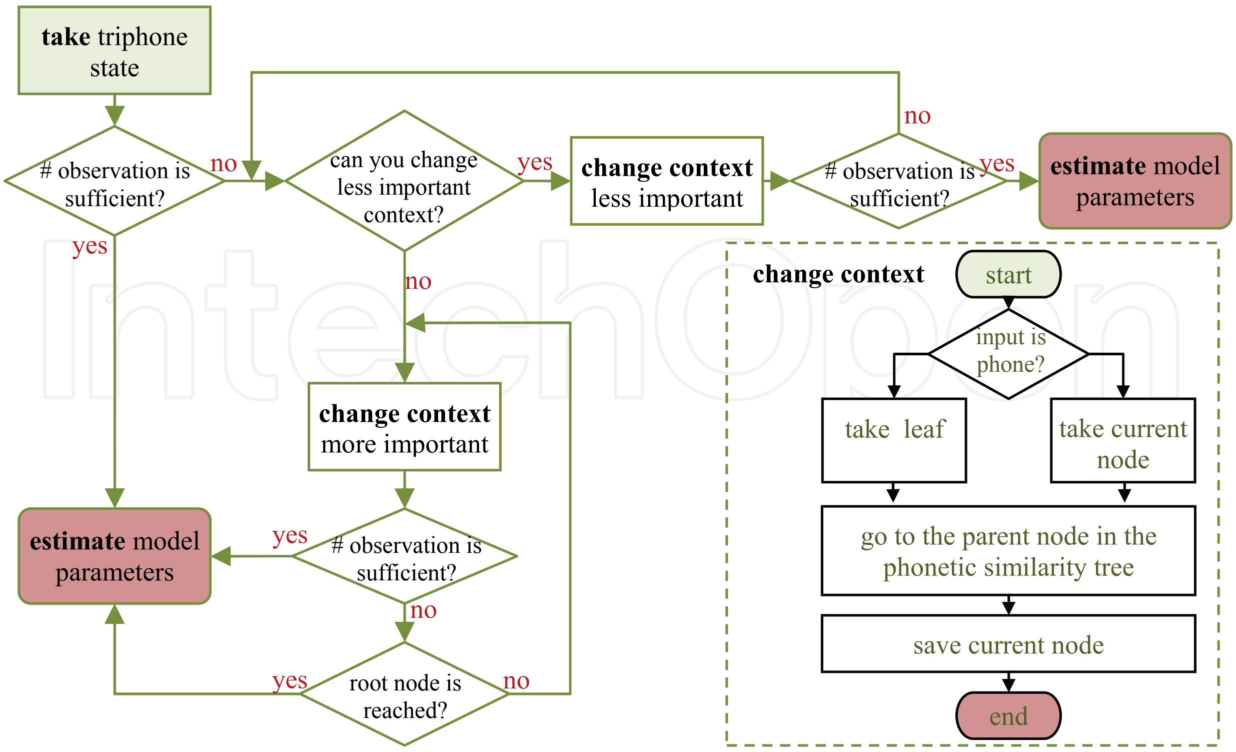

take triphone state
# observation is sufficient?
no
yes
can you change less important context?
yes
no
change context less important
# observation is sufficient?
no
yes
estimate model parameters
change context more important
# observation is sufficient?
yes
no
estimate model parameters
root node is reached?
yes
no
change context
start
input is phone?
take leaf
take current node
go to the parent node in the phonetic similarity tree
save current node
end

Fig. 6. Flowchart of the tying procedure.

modelling the phone *Ph* being in different contexts. The algorithm starts with the states whose more important context is *MIC* and the less important context is any phone in parent node of the phonetic similarity tree for the phone *LIC*. If in this attempt the sufficient number of observations is not obtained, the algorithm extends the search to states belonging to the *i*-th state of the phone *Ph* whose more important context is *MIC* and less important context is any phone contained by one step higher parent node containing the phone *LIC*. The previous step is repeated until the sufficient number of observations is obtained or the root node is reached. If the root node is reached and a sufficient number of observations is not, then the algorithm tries to borrow additional observations from the *i*-th state of the phone *Ph*, whose the less important context is arbitrary and the more important context is any phone in the parent node containing phone *MIC*. If in this attempt a sufficient number of observations is not obtained, the algorithm extends the search on states, which belong to the *i*-th state of the phone *Ph* whose less important context is arbitrary and more important context is any phone in the one step higher parent node containing phone *MIC*. The previous step is repeated until a sufficient number of observations is obtained or the root node is reached (Delić at al., 2007).

3.4 Vocal tract length normalisation

Acoustic variations between training and test conditions, caused by different microphones, channels, background noise as well as speakers, are known to deteriorate ASR performance. Variations caused by speakers can be divided into extrinsic and intrinsic. Extrinsic variations are related to cultural variations among speakers as well as their emotional state, resulting in diverse speech prosody features. Intrinsic variations are related to speaker anatomy (vocal tract dimensions).

The state-of-the-art ASR systems based on HMM and GMM are sensitive to differences in training and test conditions, which result in serious degradations of performance (Molau, 2003; Benzeghiba et al., 2006). One of the common methods to reduce spectral variations caused by different vocal tract length and shape is vocal tract length normalisation (VTN). There are several algorithms proposed in the literature. There are two approaches based on: *i*) formant position and *ii*) maximum likelihood criterion. The goal of the algorithms based on formant position is to find spectrum frequency warping function which map average (sample mean or median) formant position of some speaker into average formant position of universal speaker (Gouvea & Stern, 1997; Jakovljević et al., 2006). On the other hand, the goal of the algorithms based on the maximum likelihood criterion is to find spectrum frequency warping function, which transforms feature vectors of some speaker on the way which leads to increased theirs likelihood on the universal speaker model (Lee & Rose, 1996; Welling at al., 1999). Modification of this approach is presented in (Miguel et al., 2008) where this transformation is incorporated into a so called 2-D HMM model.
The work presented in this chapter is based on (Welling et al., 1999). Piecewise linear spectrum warping function is chosen as the most effective one and its implementation the simplest one.
It is defined as:

$$\omega_\alpha = \begin{cases} \alpha\omega & \omega \le 7\pi/8 \\ \alpha\omega - (8-7\alpha)(\omega - 7\pi/8) & 7\pi/8 \le \omega \le \pi \end{cases} \tag{4}$$

where ω is the original frequency and ω_a scaled frequency and a VTN coefficient. In order to reduce search space, VTN coefficients are discrete and usually take values from 0.88 up to 1.12 with step 0.02.
The criterion to choose VTN coefficient is:

$$\alpha_r = \arg\max_\alpha P(X_{r,\alpha} \mid W_r; \lambda_k) \tag{5}$$

where $X_{r,a}$ are all feature vectors which belong to the speaker r normalised by the VTN coefficient a, and W_r are the corresponding transcriptions, and λ_k model of the universal speaker.
The training procedure can be summarised into two steps:

1. VTN coefficient estimation for each speaker in the training phase;
2. Training of HMM models which will be used in the recognition process.

Additionally, the test procedure basing on a multiple pass strategy includes three steps:

1. Initial recognition of the original (unnormalised) sequence of the feature vectors using a speaker independent model set. The output consists of initial transcription and phoneme boundaries;
2. VTN estimation using initial transcription generated in the previous step. The procedures of VTN coefficient estimation are the same as those in the training process. Note that estimation of VTN coefficients in the test procedure is burdened with additional uncertainty because initial transcriptions and phone boundaries can be incorrect (which is not the case in the training phase);
3. Final recognition of the sequence of feature vectors normalised by the VTN coefficient estimated in the previous step. The VTN coefficients are estimated by using a speaker independent ASR system trained on the normalised features.

The models with one Gaussian per HMM state are chosen as models for VTN estimation, because of their general nature and the fact that they do not adapt to the features of a particular speaker, unlike HMM models with more than one Gaussian mixture per state (Welling et al., 1999).

We claim that the disadvantage of the standard procedure for VTN coefficient estimation defined by (5) is it's favouring of longer and more frequent phonemes (their frames are dominant in likelihood estimation of the sequence). Here we suggest several optional criteria. For the sake of convenience the method described by (5) in the further text will be referred to as M0.

In order to eliminate the influence of phone duration on VTN coefficient estimation, the value which maximises average likelihood per phone instance should be used as VTN coefficient. The term "phone instance" stands for one particular realisation of corresponding phoneme in the speech corpus. This criterion can be summarised as:

$$\alpha_r = \arg\max_{\alpha} \frac{1}{N_{pi}} \sum_{n=1}^{N_{pi}} P_n(X_{n,r,\alpha} \,|\, W_n; \lambda_k) \tag{6}$$

where $P_n(X_{n,r,a} | W_n; \lambda_k)$ is the likelihood of the phone instance W_n on the universal model set λ_k and the observations belonging to the given phone instance $X_{n,r,a}$, N_{pi} is the number of the all phone instances belonging to the speaker *r*. The scaling factor $1/N_{pi}$ is not essential, but for comparison of the average values between different speakers it is. The likelihood of the phone instance can be calculated as sample mean or sample median of the likelihoods of the observations belonging to the phone instance. The first variant in the further text will be referred to as M1 and the second as M2. Favouring phonemes with more instances in the corpus was motivated by the idea to choose a VTN which results in higher likelihood for a larger number of phone instances, and in vowels as most frequent phonemes. The weakness of this method is that it does not result in the optimal increase of word sequence likelihood, since phone instances of longer durations have greater influence than phone instances of shorter durations. Note that the goal of training and test (decoding) procedure is to obtain the maximum likelihood of word sequence. The motivation for M2 method is similar to the one for the M1 method, with an additional aim of experimenting with robust methods for estimation of likelihood of phone instances. With the use of sample median instead of sample mean the influence of extremely low and high values of feature vector likelihood is eliminated.

In order to eliminate the influence of phone duration and frequency in VTN coefficient estimation, the value which maximises average likelihood per phoneme should be used as the VTN coefficient. The likelihood per phoneme represents the average of the likelihoods of all feature vectors belonging to the given phoneme. We proposed four variants which differ in the way how average likelihood per phone and average phone likelihood is calculated. The method, which is in further text referred to as M3, calculates both average likelihood per phoneme and average phoneme likelihood as sample mean. The method referred as M4 is similar to the M3, but it calculates average phoneme likelihood as sample median. The methods referred to as M5 and M6 are similar to the M3 and M4 respectively, but they calculate average likelihood per phoneme as sample median.

None of the methods M3-M6 results in the increase of the likelihood of word sequence. The M4 method represents a robust version of the M3 method. The explanation is the same as the one for the M2 method. The M5 and M6 methods represent robust versions of the M3 and M4 methods respectively. The use of sample median instead of sample mean results in the elimination of influence of extremely low and high values of phoneme likelihoods.

None of the proposed methods take into consideration non-speech, damaged segments and segments with occlusions of plosives and affricates. All of them use the same initial model set (with one Gaussian per state). All final model sets have the same topology i.e. the number of models, states and mixtures.

The standard features used in VTN estimation procedure are the same as the features used in the recognition process. This approach is based on the reasoning that a VTN coefficient should reduce inter-cluster variations for both static and dynamic features, although the theoretical motivation for VTN includes only spectrum envelope modifications (static features).

However, in the histogram which represents the frequency of the VTN coefficients in the training corpus, there is a significant peak at 1.04 for the female speakers, as shown in Fig. 7. The analysis of the causes which lead to the peak at 1.04 in the histogram included the analysis of the curves describing the dependency of average likelihood on VTN coefficients. These are the curves used for VTN estimation (the estimated value of a VTN coefficient is the point where the curve reaches its maximum). These curves for a majority of the female speakers with estimated VTN value equal to 1.04, are bimodal (two close local maxima, as shown in Fig. 8. a)) instead of unimodal (only one local maximum, as shown in Fig. 8. b)), the latter being expected as more common.

Excluding dynamic features from the VTN estimation procedure results in a unimodal shape of the decision curves for all speakers. The values of word error rate WER on the standard test corpus for all estimation methods are presented in Table 1. The cases when only static and both static and dynamic features are used are given in the first and second row, respectively. The results show that if dynamic features are omitted, the WER is smaller for a majority of the proposed methods of VTN estimation. In the case of M6 method, the opposite result is caused by smaller efficiency of the sample median in the test phase. The same holds for the M5 method, but the result was not contrary to the majority.

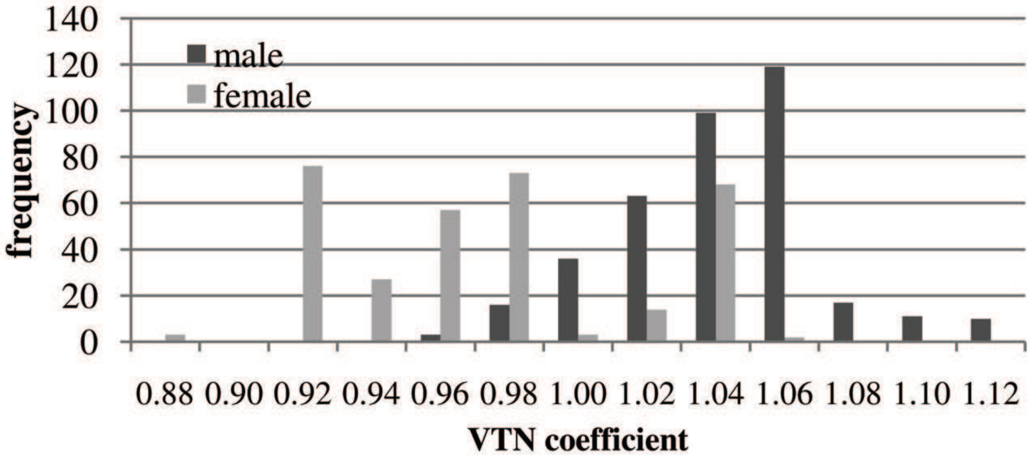
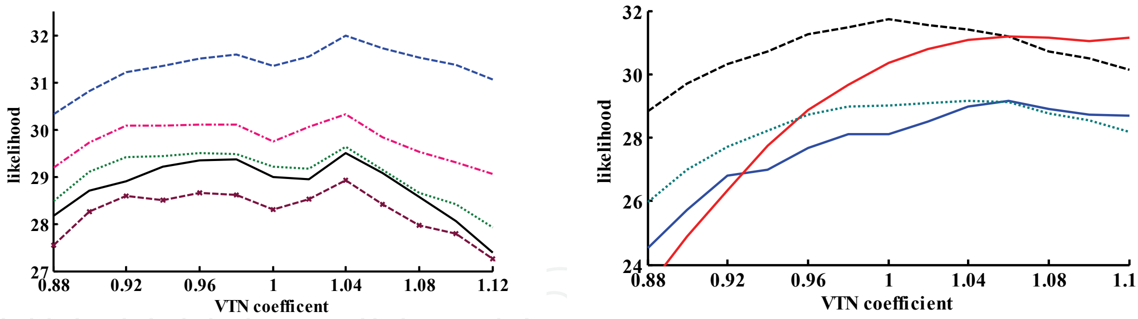


Fig. 7. The histogram of VTN coefficients for male and female speakers in the training corpus in case of M0 estimation method. For other proposed methods similar histograms are obtained.



a) b)

Fig. 8. a) The examples of the bimodal shapes of the VTN decision curves typical for the most female speakers with VTN coefficient equal to 1.04. b) The examples of unimodal shapes of the VTN decision curves typical for the majority of the speakers.

	M0	M1	M2	M3	M4	M5	M6
s	4.28	4.52	4.38	4.07	4.38	4.38	4.59
s+d	4.45	4.66	4.80	4.66	4.38	4.90	4.49

Table 1. The values of WER for the methods of VTN estimation depending on whether static or both static and dynamic features are used

	M0	M1	M2	M3	M4	M5	M6
norm.	4.28	4.52	4.38	4.07	4.38	4.38	4.59
unnorm.	5.07	5.31	5.11	4.76	4.55	5.42	4.61

Table 2. The values of WER for the methods of VTN estimation in case the HMM set is trained on normalised (norm.) or unnormalised (unnorm.) features

The motivation to explore the necessity for the iterative VTN coefficient estimation in the training phase is based on the fact that initial results showed significant differences depending on whether an HMM set, used for the VTN estimation, was trained on the normalised or on the unnormalised set of features. The results are shown in Table 2. Note that both HMM sets used in the VTN estimation procedure have the same complexity i.e. they consist of a single Gaussian density per triphone state. These differences suggest that VTN values estimated in the training phase could be improved (so as to result in a lower WER), suggesting that an iterative procedure should be adopted.

The iterative procedure can be summarised into the following three steps:

1. An HMM set λ_k, in the k-th iteration step, containing triphone states with a single Gaussian density, is trained on the feature vectors normalised by appropriate VTN coefficients for each speaker. The VTN coefficient values are in the initial step equal to 1 for all speakers and in the other steps equal to the values estimated in the previous step.
2. For each speaker in r the training corpus, a VTN coefficient a_r is chosen as the value which maximises the average likelihood per observation or phone instance or phoneme depending of method (M0-M6).
3. Repeat steps 1 and 2 until the number of changes or average change becomes sufficiently small. In this paper, the stopping condition is satisfied when the average change of VTN coefficients becomes smaller than one half of the VTN coefficient step (i.e. 0.01).

	#sub	#ins	#del	WER[%]	RI1[%]	RI2[%]	RI3[%]
REF1	94	56	9	5.94			
REF2	94	51	8	5.28			
M0	65	44	6	3.97	27.7	24.8	11.1
M1	65	39	5	3.76	31.4	28.7	16.8
M2	60	36	5	3.49	36.5	34.0	20.3
M3	69	42	8	4.11	25.2	22.2	10.0
M4	64	46	7	4.04	26.4	23.5	11.1
M5	66	39	5	3.80	30.8	28.1	16.0
M6	69	50	9	4.42	19.5	16.3	2.2

Table 3. Performance of the analysed system and its relative improvement in comparison to three referent systems (REF1, REF2, original M0). The method M0, whose performance is shown in the table, is different from the original M0 in that it uses only static features and iterative procedure for VTN estimation.

The complete results are presented in Table 3. The first referent system (REF1) represents a speaker independent ASR system. The complexity of this system is the same as the complexity of all systems which used VTN. The second referent system (REF2) is a gender dependent ASR system, with slightly smaller complexity than the other ASR systems which are analysed. The remaining systems include VTN estimation, differing between themselves in the type of VTN estimation used. Their relative improvements (RI) in comparison to REF1, REF2 and basic M0 method proposed in (Welling et al., 1999) are presented in the last three columns of Table 3, respectively.

All VTN system results in significant RI comparing to the referent systems REF1 and REF2. VTN methods M1 and M2 achieve the best performance, but McNemar test (Gillick & Cox, 1989) shows that the differences are not statistically significant in comparison to the method M0 (only static features and iterative VTN estimation procedure), M4 and M5.

Some of the proposed VTN estimation methods results in noteworthy RI comparing to baseline VTN methods (see RI3 for M1 and M2). These differences are proved statistically significant by McNemar test. A possible explanation could be that vowels are frequent phonemes and they contain more information about vocal tract length then other phonemes. The VTN estimation methods which disregard frequency and duration of phonemes (M3-M6) demonstrate significant variations in WER depending on whether the sample mean or the median is used. These variations are probably the result of an insufficient number of instances in the test phase. The results of the experiments with fast VTN tests support the previous statement (Jakovljević, 2009). The improvement in the case of M4 and M6 is minor, which can be explained by small efficiency of sample median used for estimation of average phone likelihood on the test set.

3.5 Gaussian selection

In order to obtain a high level of accuracy, HMM based CSR systems typically use continuous densities. Most of them tend to operate several times slower than real time which eventually makes them too slow for any real–time application. In such systems, calculation of state likelihoods makes up a significant proportion (between 30-70%) of the computational load. Actually, each state usually contains a significant number of Gaussian components in the corresponding mixture that are all separately evaluated in order to determine the overall state likelihood. Many techniques could be applied in order to reduce the computations required. Some of them target dimensionality reduction (like linear discriminant analysis or heteroscedastic linear discriminant analysis), some of them tying of acoustical states (semi-continuous HMM models), and there is also a number of fast Gaussian Selection (GS) methods that for each frame obtain the desired set of baseline Gaussians to be calculated exactly, based on a pre defined data structure. Of course, the goal is to increase the speed of speech recognition system without degrading the recognition accuracy. There are two distinct classes of GS methods: bucket box intersection (Woszczyna et al., 1997) and clustering (Bocchieri, 1993), (Knill et al., 1996), (Knill et al., 1999). We developed our own GS method, which is described in detail in (Janev et al., 2008).

The basic idea behind the clustering GS method is to form hyper-mixtures by clustering close baseline Gaussian components into a single group (clusters) by means of Vector Quantisation (VQ) assigning to each cluster unique hyper-density (almost always Gaussian) with parameters estimated in the appropriate way. In the decoding process, only those baseline Gaussian components belonging to clusters with corresponding hyper-densities whose "distance" to the particular speech frame is above predefined threshold are calculated directly, while the likelihood of others are floored with some approximate values. It significantly improves computational efficiency with relatively small degradation in recognition performances (Janev et al., 2008). There is no problem if the overlaps between Gaussian components are small, and their variances are of the same range. However, in real case, there are numerous models which do not fit this profile. Actually, significant overlapping between Gaussian components is common situation in CSR systems.

Baseline VQ based Gaussian selection is based on (Bocchieri, 1993). Actually, during the training phase the acoustical space is divided up into a set of VQ regions. Each Gaussian component (mixture) is then assigned to one or more VQ codewords (VQ Gaussian mixture clustering). During the recognition phase, the input feature vector is vector quantised, i.e. the vector is mapped to a single VQ codeword. The likelihood of each Gaussian component in this codeword shortlist is computed exactly, whereas for the remaining Gaussian components the likelihood is floored i.e. approximated with some back-off value. The clustering divergence that we have used in VQ based approach was of course different than the one that used in (Bocchieri, 1993) because it is not suitable enough for application with full covariance Gaussians. It was taken from the more theoretical works presented in (Goldberg et al., 2005) and (Banerjee et al., 2005). It is the most appropriate and theoretically motivated approach for the simplification of a large Gaussian mixture (with large number of components) into smaller (Shinoda et al., 2001), (Simonin et al., 1998), which is a significant part of the problem in the GS clustering approach. It can be showed that generalised k-means clustering leads to the local minimum of the target function that represents symmetric KL divergence between the baseline Gaussian mixture *f* and its simplification *g*:

$$D\left(f\,||\,g\right)=\sum_{i=1}^{k}\alpha_i \min_{j=1}^{n} KL\left(f_i\,||\,g_j\right), \tag{7}$$

where f_i and g_j are components of mixtures *f* and *g*, and a_i is the occupance of f_i. This is actually a generalisation of the well known Lindo-Buzo-Gray algorithm (Knill et al., 1996), (Lindo et al., 1995). The algorithm actually obtains the local minimum of $D(f||g)$ by iteratively repeating REGROUP and REFIT steps. In the REGROUP step, every baseline Gaussian component θ_m is assigned to the unique cluster chosen so that the symmetric KL divergence $KL(\theta_m, \theta_f)$ to the hyper-Gaussian θ_f that corresponds to cluster is minimal. In the REFIT step, parameters of the "new" hyper-Gaussian (c_f, Σ_f) that correspond to the particular cluster are estimated in the Maximum Likelihood manner i.e. equivalently as the ones that minimise the KL divergence between the underlying Gaussian mixture that corresponds to the particular cluster and the actual hyper-Gaussian (Banerjee et al., 2005):

$$\hat{c}_f = \sum_{m=1}^{M_f} w_m \mu_m \tag{8}$$

$$\hat{\Sigma}_f = W_f + \sum_{m=1}^{M_f} w_m (\hat{\mu}_m - \hat{c}_f)(\hat{\mu}_m - \hat{c}_f)^T \tag{9}$$

$$W_f = \sum_{m=1}^{M_f} w_m \hat{\Sigma}_m \tag{10}$$

The term W_f is the pool covariance matrix of the f-th cluster, while w_m is the mixture cluster occupancy (the whole concept could be given strait forward in the terms of soft posterior probabilities obtained using Baum Welch algorithm, but are omitted for the simplicity as in (Janev et al., 2008)).

The main idea how to decrease the influence of significant overlapping of baseline Gaussians is for GS process to be driven by the eigenvalues of covariance matrices of Gaussians to be selected. The basic idea is to group the baseline Gaussian components on the basis of their eigenvalues into several groups, before the actual VQ clustering is applied on each group separately. The method is referred as Eigenvalues Driven Gaussian Selection (EDGS). If the baseline VQ clustering is performed on the whole set of Gaussian components, then at the end of the procedure, in some cluster, there could be both components for which the eigenvalues of covariance matrices are predominantly large, and those for which the eigenvalues of covariance matrices are predominantly small. This is especially the case if the degree of Gaussian components overlapping is high, because many low-variance mixtures could be masked by high-variance ones and thus assigned to the same cluster. This comes as a consequence of the use of symmetric KL clustering distance, more precisely, its Mahalanobis component. As a result, the covariance matrix of the hyper-Gaussian that corresponds to a cluster can have predominantly large eigenvalues, although there are many baseline Gaussian components belonging to that cluster with predominantly small eigenvalues of covariance matrices.

Baseline Gaussian components are masked by high-variance ("wide") ones, thus in the decoding process the following can happen. If the likelihood of a hyper-Gaussian evaluated on the input vector is above the predefined threshold, all baseline components in the cluster will be evaluated for that particular input vector.

The performance of a Gaussian selection procedure is assessed in terms of both recognition performance and reduction in the number of Gaussian components calculated. Reduction is described by the computation fraction CF, given as CF = $(G_{new} + R_{comp})/G_{full}$, where G_{new} and G_{full} are the average number of Gaussians calculated per frame in the VQGS and the full system respectively, and R_{comp} is the number of computations required for the system to calculate log-likelihoods of hyper-mixtures in order to decide whether the mixtures belonging to that cluster will be evaluated or not. The evaluation will include even those mixtures with low likelihood values that should have been excluded from the evaluation in order to obtain a sufficient reduction in computational load and at the same time not to change WER significantly. The result is the increase in both CF and WER. It is essentially for EDGS to work that we keep the average number of baseline components in cluster n_{avr} reasonably small. Nevertheless, the similar constraint must also be met in order to obtain satisfactory recognition accuracy of any GS system.

As a result of situations when low-variance ("narrow") components are masked by high-variance ("wide") ones, in the decoding process the following can happen. If the likelihood of a hyper-Gaussian evaluated on the input vector is above the predefined threshold, all the baseline components in the cluster will be evaluated for that particular input vector. The evaluation will include even those components with low likelihood values that should have been excluded from the evaluation in order to obtain a sufficiently low CF and at the same time not to change WER significantly. The result is the increase in both CF and WER. Thus, EDGS proceeds with the combining of the most significant eigenvalues of the baseline Gaussian covariance matrices in order to group them in the predefined number of groups, prior to the execution of the VQ clustering on each group separately. The largest eigenvalues are the most important for mixture grouping and their relative importance decreases with their value. For the aggregation of the value on the base on which the particular Gaussian component is to be grouped, we have proposed the usage of Ordered Weighted Average OWA aggregation operators (Janev et al., 2008). The idea is to give more weight to more significant (larger) eigenvalues in the aggregation process, thus optimising the OWA weights. They are to be applied to the particular eigenvalues vector $\lambda = (\lambda_1, \ldots, \lambda_p)$ in the following way:

$$OWA_{\omega}(\lambda_1, \ldots, \lambda_p) = \sum_{j=1}^{p} \omega_j \lambda_{\sigma(j)} \tag{11}$$

where $0 \leq \lambda_{\sigma(1)} \leq \ldots \leq \lambda_{\sigma(p)}$. Depending on the OWA values, mixtures are divided into groups. The coefficients $\omega \in R^p$ satisfy the constrains that $0 \leq \omega_j \leq 1$ and they sum to one.

The OWA operators provide a parameterised family of aggregation operators which include many of the well known operators such as the maximum, the minimum, *k*-order statistics, median and the arithmetic mean. They can be seen as a parameterised way to interpolate between the minimum and the maximum value in an aggregation process. In this particular application, the applied operator should be somewhat closer to max(·) in order to favour more significant eigenvalues in the grouping process. The method to optimally obtain OWA coefficients introduced in (Yager, 1988) and used in (O'Hagan, 1988) is applied. The maxness $M(\omega) = \alpha \in [0,1]$ of the OWA operator is defined as:

$$M(\omega) = \sum_{j=1}^{p} \omega_j \frac{j-1}{p-1} \tag{12}$$

The idea is to maximise dispersion of weights $D(\omega)$ defined (O'Hagan, 1988) as

$$D(\omega) = -\sum_{j=1}^{p} \omega_j \ln(\omega_j) \tag{13}$$

thus obtaining the Constrained Nonlinear Programming (CNP) problem (O'Hagan, 1988). For finding the optimal weights ω_{opt}, any standard method can be used (Biggs, 1975), (Coleman et al., 1996). In the sequel, we give the baseline VQGS and EDGS algorithms as follows:

VQGS

Initialisation:

- For predefined n_{avr} and the overall number of mixtures M, calculate the number of clusters as: $N_{hpc} = |X| =\{M/n_{avr}\}$.
- Pick up at random (uniform distribution) N_{hpc} different centroids c_f f $\in \{1,\ldots,N_{hpc}\}$ from the set of overall M mixture centroids used. Assign to every centroid the identity covariance matrix $\Sigma_f = I$. Let Gaussian densities $X^{(0)} = \{\chi_f(c_f, \Sigma_f): f = 1,\ldots,N_{hpc}\}$ be initial hyper-mixtures.

Clustering:

Do the following, for predefined $\varepsilon > 0$

- To all mixtures θ_j, $j = 1,\ldots M$ assign a corresponding hyper-mixture $\chi^{(j)}$ in the current k-th iteration as: $\chi^{(j)}$ =argmin $d(\theta_j, \chi)$, where $d(\cdot,\cdot)$ is symmetric KL divergence.
- Evaluate hyper-mixture parameters c_f and Σ_f using ML estimates (8), (9) and (10), to obtain $\mathrm{X}^{(k)}$
- If any cluster "runs out" of mixtures, set $N_{hpc} = N_{hpc}$ - C for the next iteration, where C is the number of such clusters.

Until $D_{average} < \varepsilon$, for $D_{average}$ defined by (7).

EDGS:

Initialisation:

- Specify the number of groups G.
- Using any CNP method, obtain optimal OWE weights for predefined maxness $a\in[0,1]$ as: ω_{opt} = argmax $D(\omega)$, satisfying constraints $M(\omega) = a$, that $0 \le \omega_j \le 1$ and they sum to one.
- For ω_{opt}, determine the group threshold vector (elements are group borders) $\tau = [\tau_{\max}{}^{(1)},\ldots \tau_{\max}{}^{(G-1)}]$, and set $\tau_{\min}{}^{(g+1)} = 0$, $\tau_{\max}{}^{(g)} = = \infty$. The group borders should satisfy the constraint: $\tau_{\max}{}^{(g+1)} = \tau_{\max}{}^{(g)}$, for $g = 1,\ldots$,G-2, where $\tau_{\max}{}^{(1)}$ is obtained heuristically.

Mixture Grouping:

For every $i = 1,\ldots,M$, for mixture θ_i do:

- Obtain eigenvalues $\lambda^{(i)} = (\lambda_1{}^{(i)},\ldots, \lambda_p{}^{(i)})$.
- Assign θ_i to the group giff: $OWE\omega_{opt}(\lambda^{(i)}) \in [\tau_{\min}{}^{(g)}, \tau_{\max}{}^{(g)})$

Perform baseline VQGS method on every group separately to obtain clusters with mixtures and corresponding hyper-mixtures.

The decoding process is given as follows

Decoding:

For all observations x_t, $t = 1,\ldots,N$, where N is the number of observations in the testing process do for every cluster C_k, k=1,…,N_{hpc} do:

- Evaluate log-likelihood $\ln f(x_t, \chi^{(k)})$, where $\chi^{(k)}$ is the hyper-mixture that corresponds to cluster C_k.
- If $\ln f(x_t, \chi^{(k)}) > \theta$, where θ is a predefined likelihood threshold, evaluate the exact likelihood for all mixtures that belong to the cluster C_k. Else, set all belonging mixture log-likelihoods to $\ln f(x_t, \Theta^{(k)})$ where $\Theta^{(k)}$) is the Gaussian mixture with centroid c_k and covariance matrix W_k defined by (10).

5. Conclusion

Both ASR and TTS systems described in this chapter have been originally developed for the Serbian language. However, linguistic similarities among South Slavic languages have allowed the adaptation of this system to other South Slavic languages, with various degrees of intervention needed.

As for ASR, adaptation to Bosnian and Croatian was very simple (due to extreme similarity of phonetics), whereas for Macedonian it was necessary to develop separate speech databases. The actual procedures used for ASR were almost identical in all cases. While well known algorithms were used for model training and testing, in this chapter only the original algorithms are presented. The VTN procedure based on the use of the iterative method and only static features for VTN coefficient estimation shows significant improvement in comparison to the common VTN procedure. The eigenvalue driven Gaussian selection significantly reduce computational load with minor increase of WER. Neither of the proposed algorithms is language dependent.

As for TTS, conversion of an arbitrary text into intelligible and natural-sounding speech has proven to be a highly language-dependent task, and the degree of intervention was variable and depended on specific properties of a particular language. For example, the simplicity of accentuation in Macedonian has allowed POS tagging and syntactic parsing to be avoided altogether, at the price of certain impairment in quality of synthesis. On the other hand, for Croatian and Bosnian, it was also necessary to build new accentuation dictionaries and to revise the expert system for POS tagging in order to assign words their appropriate accentuation, necessary for production of natural sounding speech.

It can be concluded that, in spite of the apparent language dependence of both principal speech technologies, some of their segments can be developed in parallel or re-used. The ASR and TTS systems described here are widely applied across the Western Balkans. In fact, practically all applications of speech technologies in the countries of the Western Balkans (Pekar et al., 2010) are based on ASR and TTS components described in this chapter.

5.1 Directions for future work

The team at the University of Novi Sad is a core of a greater multidisciplinary team in Serbia, whose aim is to further increase the quality of synthesised speech and the accuracy and robustness of ASR. The ultimate goal is to incorporate ASR and TTS into (multimodal) spoken dialogue systems, to expand ASR to larger vocabularies and spontaneous speech, not only in Serbian but in other South Slavic languages as well. Development of speech technologies for a language represents a contribution to the preservation of the language, overcoming language barriers and exploiting all the benefits coming from the use of speech technologies in one's native language.

6. References

Banerjee, A.; Merugu, S.; Dhillon, I. & Ghosh, J. (2005). Clustering with Bergman divergence, *Journal of Machine Learning Research*, Vol 6, pp. 1705-1749

Beutnagel, M.; Mohri, M. & Riley, M. (1999). Rapid unit selection from a large speech corpus for concatenative speech synthesis, *Proceedings of 6th EUROSPEECH*, pp. 607-610, ISSN 1018-4074, Budapest, Hungary

Benzeghiba, M.; De Mori R.; Deroo, O.; Dupont, S.; Jouvet, D.; Fissore, L.; Laface, P.; Mertins, A.; Ris, C.; Rose, R.; Tyagi, V. & Wellekens, C. (2006). Impact of Variabilities on Speech Recoginition, *Proceedings of 11th SPECOM* (*Speech and Computer*), St. Petersburg, Russia

Biggs, M. (1975). Constrained minimization using recursive quadratic programming. *Dixon LCW, Szergo GP (Eds.) Towards global optimization*. North-Holland, Amsterdam, pp. 341–349

Bocchieri, E. (1993). Vector quantization for efficient computation of continuous density likelihoods. *Proceedings of ICASSP*, Minneapolis,MN, Vol 2, pp. II-692–II-695

Brill, E. (1992). A simple rule-based part of speech tagger. *Proceedings of the 3rd Conference on Applied Natural Language Processing*, pp. 152-155, Trento, Italy

Coleman, T. & Li, Y. (1996). An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM J Optim* 6, pp. 418–445

Delić, V.; Pekar, D.; Obradović, R.; Jakovljević, N. & Mišković, D. (2007). A Review of AlfaNum Continuous Automatic Speech Recognition System, *Proceedings of 12th SPECOM (Speech and Computer)*, pp. 702-707, ISBN 6-7452-0110-x, Moscow, Russia, October 2007

Delić, V. (2000). Speech corpora in Serbian recorded as a part of AlfaNum project, *Proceedings of 3th DOGS (Digital Speech and Image Processing),* pp. 29-32, Novi Sad, Serbia, October 2000, Novi Sad

Dutoit, T. (1997). *An Introduction to Text-to-Speech Synthesis*, Kluwer Academic Publishers, ISBN: 0-7923-4498-7, Dordrecht/Boston/London

Hajič, J. & Hladká, B. (1998). Czech language processing – POS tagging. *Proceedings of 1st International Conference on Language Resources and Evaluation*, pp. 931-936, Granada, Spain

Gillick, S. & Cox, S. (1989). Some Statistical Issues in the Comparison of Speech Recognition Algorithms, *Proceedings ICASSP,* pp. 532-535

Goldberg, J. & Roweis, S. (2005). Hierarchical clustering of a mixture model, *Proceedings of NIPS 2005*, December 5, Vancouver

Gouvea, E. & Stern, R. (1997). Speaker Normalisation through Formant Based Warping of Frequency Scale, *Proceedings of EUROSPEECH*, pp. 1139-1142, Rhodes, Greece

Jakovljević, N.; Mišković, D.; Sečujski, M. & Pekar, D. (2006). Vocal Tract Normalisation Based on Formant Positions, *Proceedings of IS-LTC*, Ljubljana, Slovenia

Jakovljević, N.; Sečujski, M. & Delić, V. (2009). Vocal Tract Length Normalisation Strategy Based On Maximum Likelihood Criterion, *Proceedings of EUROCON*, pp. 417-420, ISBN 978-1-4244-3861-7, St. Peterburg, Russia

Jakovljević, N. (2009). *Improvement of ASR performance using Vocal Tract Length Normalisation* (*M.Sc. thesis)*, Faculty of Technical Sciences, University of Novi Sad, Serbia (in Serbian)

Janev, M.; Pekar, D.; Jakovljević, N. & Delić, V. (2008). Eigenvalues driven gaussian selection in continuous speech recognition using HMMs with full covariance matrices. *Applied Intelligence, Springer Netherlands,* DOI: 10.1007/s10489-008-0152-9, (Print, accepted) December 2008, ISSN 0924-669X, 1573-7497 (Online, available) http://www.springerlink.com/content/964vx4055k424114/

Jurafsky, D. & Martin, H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice-Hall, ISBN-10: 0131873210, Upper Saddle River, NJ.

Knill, M.; Gales, F. & Young J. (1996). Use of Gaussian selection in large vocabulary continuous speech recognition using HMMs, *Proceedings of Int. Conf. Spoken Language Processing*

Knill, M.; Gales, F. & Young, J. (1999). State based Gaussian selection in large vocabulary continuous speech recognition using HMMs, Mar 1999, Vol 7, Issue 2, pp. 152-161

Lee, L. & Rose, R. (1996). Speaker Normalisation using Efficient Frequency Warping Procedures, *Proceedings of ICASSP*, pp. 353-356

Lindo, Y.; Buzo, A. & Gray, R. (1980). An algorithm for vector quantizer design. *IEEE Trans Commun COMM* 28, pp. 84–95

Miguel, A.; Lleida, E.; Rose, R.; Buera, L.; Saz, O. & Ortega, A. (2008). Capturing Local Variability for Speaker Normalisation in Speech Recognition, *IEEE Transactions on Audio, Speech and Language Processing*, pp. 578-593

Molau, S. (2003) Normalisation of Acoustic Feature Space for Improved Speech Recognition, (PhD Thesis), RWTH Aachen, Germany

Obradović, R. & Pekar, D. (2000). C++ Library for Signal Processing. *Proceedings of DOGS (Digital Speech and Image Processing)*, Novi Sad, Serbia, pp. 67-70.

O'Hagan, M. (1988). Aggregating template or rule antecedents in real time expert systems with fuzzy set logic. *Proceedings* of the 22-th annual IEEE Asilomar conferences on signals, systems and computers, Pacific Grove, pp. 681–689

Oravecz, C. & Dienes, P. (2002). Efficient stochastic part-of-speech tagging for Hungarian. *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pp. 710-717, Las Palmas, Spain

Riley, M. D. (1989). Some applications of tree-based modeling to speech and language indexing. *Proceedings of the DARPA Speech and Natural Language Workshop*, pp. 339-352. Morgan Kaufmann

Sečujski, M. (2002). Accentuation dictionary of Serbian language intended for text-to-speech synthesis (in Serbian), *Proceedings of 4th DOGS (Digital Speech and Image Processing)*, pp. 17-20, Bečej, Serbia, May 2002, Publisher: FTN Novi Sad

Sečujski, M.; Obradović, R.; Pekar, D.; Jovanov, Lj. & Delić, V. (2002). AlfaNum System for Speech Synthesis in Serbian Language. *Proceedings of TSD (Text, Speech and Dialogue)*, pp. 237-244, ISBN 3-540-44129-8, Brno, Czech Republic, September 2002. *Lecture Notes in Artificial Intelligence*, *Springer-Verlag Berlin Heidelberg*, LNAI 2448, pp. 237-244, ISSN 0302-9743

Sečujski, M. (2005). Obtaining Prosodic Information from Text in Serbian Language, *Proceedings of EUROCON*, pp. 1654-1657, ISBN 86-7466-218-8 (AM), Belgrade, Serbia, November 2005

Sečujski, M.; Delić, V.; Pekar, D.; Obradović, R. & Knežević, D. (2007). An Overview of the AlfaNum Text-to-Speech Synthesis System, *Proceedings of 12th SPECOM (Speech and Computer)*, pp. Ad.Vol. 3-7, ISBN 6-7452-0110-x, Moscow, Russia, October 2007

Sečujski, M. (2009). *Automatic Part-of-Speech Tagging of Texts in Serbian Language* (*PhD thesis)*, Faculty of technical Sciences, University of Novi Sad, Serbia

Sepesy Maučec, M.; Rotovnik, T. & Zemljak, M. (2003). Modelling Highly Inflected Slovenian Language. *International Journal of Speech Technology, Springer, the Netherlands,* Vol. 6, No. 3, pp. 245-257, ISSN 1381-2416

Shinoda, K. & Lee, C. (2001). A structural Bayes approach to speaker adaptation. *IEEE Trans Speech Audio Process* 9(3), pp. 276–287

Simonin, J.; Delphin, L. & Damnati, G. (1998). Gaussian density tree structure in a multi-Gaussian HMM based speech recognition system. *Proceedings of 5th Int. Conf on Spoken Language Processing*, Sydney, Australia

Tokuda, K.; Yoshimura, T.; Masuko, T.; Kobayashi, T. & Kitamura, T. (2000). Speech parameter generation algorithms for HMM-based speech synthesis, *Proc. of ICASSP*, pp. 1315-1318, Istanbul, Turkey

Uebel, L. & Woodland, P. (1999). An Investigation into Vocal Tract Length Normalisation, *Proceedings of EUROSPEECH*, pp. 2527-2530

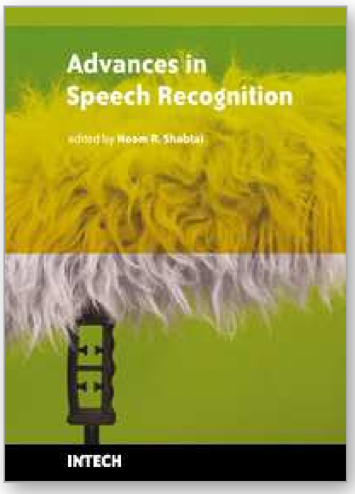
Webb, A. (1999). *Statistical Pattern Recognition*, Oxford University Press Inc, ISBN 0-340-74164-3, New York, USA

Welling, L.; Kanthak, S. & Ney, H. (1999). Improved Methods for Vocal Tract Normalisation, *Proceedings of ICASSP,* pp. 761–764, Phoenix, USA

Woszczyna, M. & Fritsch, J. (1997). Codebuch übergreifende bucket-boxintersection zur schnellen Berechnung von Emissionswahrscheinlichkeiten im Karlsruher VM-Erkenner. Verbmobil

Yager, R. (1988). On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Trans Syst Man Cybern* 18, pp. 183–190

Young, S.; Odell, J. & Woodland, P. (1994). Tree-based state tying for high accuracy acoustic modelling, *Proceedings of the workshop on Human Language Technology*, pp. 307-312, Association for Computational Linguistics, ISBN:1-55860-357-3, Plainsboro, NJ.



Advances in Speech Recognition
Edited by Noam Shabtai

ISBN 978-953-307-097-1
Hard cover, 164 pages
Publisher Sciyo
Published online 16, August, 2010
Published in print edition August, 2010

In the last decade, further applications of speech processing were developed, such as speaker recognition, human-machine interaction, non-English speech recognition, and non-native English speech recognition. This book addresses a few of these applications. Furthermore, major challenges that were typically ignored in previous speech recognition research, such as noise and reverberation, appear repeatedly in recent papers. I would like to sincerely thank the contributing authors, for their effort to bring their insights and perspectives on current open questions in speech recognition research.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821