DOI 10.15622/sp.58.3

E. PAKOCI, B. POPOVIĆ, D. PEKAR

IMPROVEMENTS IN SERBIAN SPEECH RECOGNITION USING SEQUENCE-TRAINED DEEP NEURAL NETWORKS

Pakoci E., Popović B., Pekar D. **Improvements in Serbian Speech Recognition using Sequence-Trained Deep Neural Networks.**

Abstract. This paper presents the recent improvements in Serbian speech recognition that were obtained by using contemporary deep neural networks based on sequence-discriminative training to train robust acoustic models. More specifically, several variants of the new large vocabulary continuous speech recognition (LVCSR) system are described, all based on the lattice-free version of the maximum mutual information (LF-MMI) training criterion. The parameters of the system were varied to achieve best possible word error rate (WER) and character error rate (CER), using the largest speech database for Serbian in existence and the best *n*-gram based language model made for general purposes. In addition to tuning the neural network itself (its layers, complexity, layer splicing etc.) other language-specific optimizations were explored, such as the usage of accent-specific vowel phoneme models, and its combination with pitch features to produce the best possible results. Finally, speech database tuning was tested as well. Artificial database expansion was made by modifying speech speed in utterances, as well as volume scaling in an attempt to improve speech variability.

The results showed that 8-layer deep neural network with 625-neuron layers works best in the given environment, without the need for speech database augmentation or volume adjustments, and that pitch features in combination with the introduction of accented vowel models provide the best performance out of all experiments.

Keywords: deep neural network, automatic speech recognition, chain training, LF-MMI, accents, pitch, Serbian.

1. Introduction. This paper represents an overview of results and improvements in automatic speech recognition with systems trained on the largest Serbian speech database using an effective contemporary deep neural network (DNN) architecture. Previously, there have been several experiments with a few different neural network based, as well as Gaussian mixture model (GMM) based architectures. These are mostly systems trained on smaller speech databases consisting of telephone recordings with limited spectral range, and they were tested on smaller vocabularies (up to around 14000 words) accordingly [1-2]. They are based on the cross-entropy classification criterion. On the other hand, the system in [3] was trained on the same speech database used in this paper, so there is a possibility of direct comparison. That system had input alignments from a speaker adaptive training (SAT) stage [4], and used modified stochastic gradient descent (SGD) optimization and parameter averaging [5] to compute DNN parameter values in a given number of training epochs.

In contrast to the previous methods, recently there has been a lot of talk about connectionist temporal classification (CTC) [6] in speech recognition [7], especially when there is a greater amount of data available.

CTC can also be used in the context of maximum mutual information (MMI) based sequence training, as both of them maximize the conditional likelihood of correct transcriptions. As seen in [8], and implemented in the system from this paper, some of the ideas can be applied to MMI, such as training from scratch (without initialization), a 3-fold reduced frame rate [9] using a simpler hidden Markov model (HMM) topology, and the usage of finite state acceptors (FSAs) to limit the range of frames where supervision labels can appear [10]. The proposed method is denominator-lattice-free, and the summations are done over all possible label sequences — to accomplish such a task, it is run on the GPU with a phoneme-level language model, while also using several regularization techniques to prevent, or at least reduce the possibility of overfitting.

The rest of the article is organized as follows: Section 2 describes the baseline system upon which the new system is built. Section 3 explains the training method in detail. Section 4 describes the speech database used for training, and Section 5 the language model used for decoding the test set. Section 6 briefly overviews the experiments performed, and Section 7 presents all the results in details. The following Section 8 discusses possible upgrades to the current system. Finally, Section 9 concludes the paper.

2. The baseline system. The baseline HMM-GMM speech recognition system was trained using the Kaldi speech recognition toolkit [11], which allows a long list of options for pre-DNN trainings, to provide as good as possible input alignments for the neural network. These, among others, include context-dependent triphone acoustic model training and speaker adaptive training, which were used for the baseline system here. After an initial monophone training with 1000 Gaussians as the goal, two rounds of triphone training (*tri1* and *tri2a*) were performed to eventually create a system with 3000 clustered HMM states and 25000 Gaussians in total. Alignments of the training database obtained from this system were used for SAT training (with unchanged number of states and Gaussians). Finally, the SAT system created the input alignments for DNN training. The results of decoding the test dataset (more on it in Section 4) with pre-DNN acoustic models are given in Figure 1.

3. The training method. The so-called “chain” training method is based on performing maximum mutual information (MMI) training [12-13] directly on the GPU, for the sake of benefitting from synchronized memory access across its cores, while not using lattices and implementing the forward-backward training algorithm in both the numerator and the denominator part of the objective function. Both the utterance-specific numerator graph and the shared denominator graph (which encodes all possible work sequences in the given setup) are stored as finite state acceptors (FSAs), which can be viewed as an equivalent to HMMs, but with labels instead of states on their arcs. The obvious downside of this approach is computational complexity and efficiency, so the graphs have to be simplified, to be as small as possible for all the necessary computations.

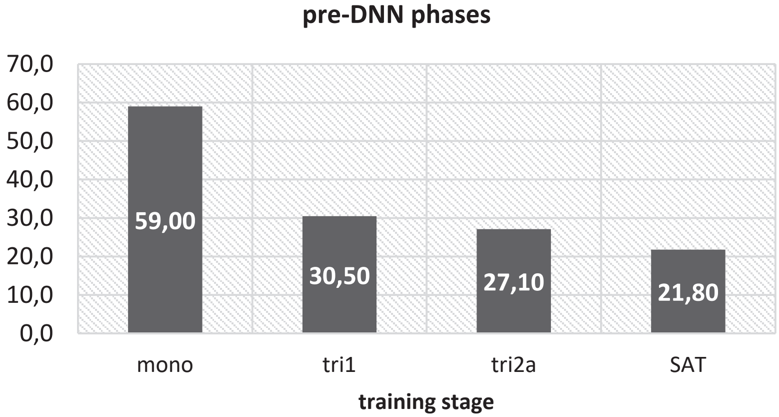

Fig. 1. Baseline WER results

Firstly, the phoneme topology is very simplified — instead of the standard left-to-right HMM topology with 3 states used in automatic speech recognition for a given phoneme model, this topology can be traversed in a single frame. Additionally, the first frame of a phoneme has a different label (the so-called "*pdf*" identifier, the neural network output) than the remaining ones, so the possible emitted sequences from one phoneme HMM are something like *a*, *ab*, *abb*, *abbb* (where *b* is a label analogous to the blank symbol in CTC) etc. A new context decision tree has to be built particularly for this new topology and the three-fold reduced frame rate (using the converted phoneme-level alignments). Furthermore, transition probabilities of HMMs are set to a constant value (0.5) since they are not that important when taking into account the presented topology.

Another simplification is in the language model used to create the denominator graph. In this method, it is a *4*-gram phoneme-level language model, estimated directly from input phoneme-level alignments of the training dataset. Conventional language models (even unigrams that are often used in MMI and similar discriminative training methods) would be way too slow to use here. Furthermore, this language model has no smoothing or pruning below trigram level to limit the graph size increase after adding context dependency, and there is a predefined total number of *4*-gram history states, selected in such a way to maximize the likelihood of training data. Finally, there is no interpolation or backoff from existing history states (test data perplexity would be infinite).

The creation of both graphs is performed in an unusual way. For the denominator, after composing with C (context dependency) and then H (HMM topology), and epsilon removal afterwards (so far like the usual HCLG graph creation in Kaldi), instead of standard determinization and minimization, a different procedure is implemented to even further reduce the number of states and transitions in the output graph — this procedure can be summarized as the sequence of operations {pushing weights; minimization; reversing the FSA; pushing weights, minimization; reversing} repeated 3 times, followed by a final epsilon removal, because the reversals can create them. The denominator graph created this way will cut down both memory consumption (less states) and time taken for MMI training (less transitions).

Moreover, the initial and final probabilities in the graph are modified. Instead of reflecting only the sentence starts' and sentence endings' statistics (which does not work with parallel training on fixed size chunks of utterances), initial probabilities are here obtained by averaging the HMM state distribution for 100 subsampled frames from the initial state, while the final probabilities are fixed to be 1.0.

As for the numerator, for a given utterance, input alignments are converted into lattices representing all possible alternative utterance pronunciations, which are then processed into phoneme graphs, and then compiled into utterance FSAs. These are processed even further [8], until each FSA state can be identified with a frame index (important for the ability to separate FSA into chunks). The numerator FSA now contains a subset of paths contained in the denominator FSA.

Like for the denominator, for processing fixed sized utterance chunks, time constraints had to be added to alignments so that it can be possible to split up the numerator FSA accordingly. Using an idea similar to one used in CTC training [12], this FSA is composed with another FSA with *number_of_subsampled_frames+1* states which has a transition from state *t* to state *t+1* with a *pdf* identifier as a label, only if that *pdf* corresponds to a phoneme that is allowed on the subsampled frame *t*. Phoneme allowance on a certain timestamp is determined based on a tolerance window (50ms), which for each phoneme in the utterance lattice allows it to appear slightly before, or slightly after from where it actually appeared.

Finally, the numerator FSA is also composed with the so-called *normalization FSA*, which is identical to the denominator FSA, but with the modified initial and final costs mentioned before. The new initial probabilities are added into the original denominator FSA using epsilon transitions from a new initial state (those epsilons are later removed).

The simpler forward-backward computation — for the numerator, is implemented on CPU, while denominator computations run on the more powerful GPU.

To reduce the possibility of overfitting [14], three different regularization methods are used — cross-entropy regularization (an additional special output layer for training the cross-entropy objective, with tweaks to the last hidden layer as well) and output l_2-norm regularization (on the main output layer), as well as the so-called leaky HMMs (allowing transitions from each HMM state into every other state, with a small coefficient, which makes the system gradually forget context).

The chosen acoustic models are sub-sampled time-delay neural networks (TDNNs), which are trained using cross-entropy training. A special set of layer splicing indexes are in use. They are *-1,0,1* for several initial layers (they see 3 consecutive frames), and *-3,0,3* for the remaining hidden layers (they see 3 frames as well, but separated by 3 frames from each other). In such a configuration, the most hidden layers need to be evaluated only on every third frame. The number of layers, number of neurons in each hidden layer, number of training epochs and other parameters, such as the coefficients for speed perturbation and volume scaling, were varied from experiment to experiment. Also, online-calculated *i-vectors* are used for the adaptation of the deep neural network along the way (with updates on every tenth frame).

4. Serbian speech database. For all the experiments, the largest Serbian speech database for LVCSR in existence up to this day was used. It is comprised of two very distinct parts — a larger part containing audio book recordings [15], read by professional speakers in a studio environment, which produced generally very high quality audio, and a smaller part containing mobile phone recordings of different people, mainly commands, inquiries and similarly structured short utterances that can be expected in a conversation with voice assistant type applications installed on mobile phones.

Naturally, the audio book part, which contains most of the material, brings a lot of variability in terms of expressiveness and the number of different sentence structures, even though the literary functional style dominates all other styles — this style is nevertheless the one most correlated with natural, everyday speech. The vast majority of the total of 121000 different words came from this part of the speech corpus. The utterances are very long on average (around 15 words per utterance), and the amount of material isn't equally distributed per speaker — some speakers have several hours of audio data, and others half an hour or even less. In the future, an equalized version of this database is going to be examined for acoustic model training. The equalization in this context implies dividing speakers with more abundant audio material into more or less equally long segments, and then modifying tempo and pitch characteristics in individual segments to create different “artificial” speakers, equally represented in the speech database as a whole. There is no significant background noise, and words and phonemes are generally well pronounced throughout the database. Everything was manually reviewed multiple times before these experiments. All in all, there is around 154 hours of audio data, out of which around 129 hours is pure speech, and the remaining 25 hours correspond to silence segments. The data is divided into more than 87000 separate utterances. In total, there are 21 identified male speakers and 27 identified female speakers, with another 10-15 different unidentified speakers (with possible overlaps).

On the other hand, the so-called “mobile” speech database consists of mostly domain-oriented utterances, as mentioned above. These utterances are much shorter (between 4 and 5 words on average), and most of them are commands, questions, numbers, currencies, proper nouns (names, cities, rivers and other topological data), different inquiries and similar sentence structures, recorded using a specialized application which simulates a conversation with the device, i.e., a helper application installed on it. There are also some regular declarative sentences, as well as spellings of names, organizations, brands, etc. People were instructed to try to talk as naturally as possible (to be more spontaneous). A lot of different speakers contain a similar amount of audio data, and all of them have all the given utterance types. This set makes up around 61 hours of total audio data, out of which 42 hours is speech, and 19 hours is silence. In total, there are 170 male and 181 female speakers, which adds up to around 74000 utterances. Recording quality is usually good, but several speakers have a significant amount of background noise as well.

All audio data was sampled at 16 kHz, 16 bits per sample, mono PCM. Both parts of the speech database were used in an attempt to train more robust acoustic models, well-adjusted to both shorter commands and longer, regular sentences. To summarize, around 160000 utterances and 215 hours of audio data was obtained in total (170 hours of speech without silences, see Table 1). Out of this, 18 hours of speech coming from 26 speakers is selected for the test set. Test speakers do not participate in any training, and each unique speaker is either completely used for testing, or completely for training (never for both). The selection of speakers was random, and the goal was to take around 10% of the more varied audio books database (9 speakers, 15 hours, 9000 utterances, 140000 words), and around 5% of the more uniform (vocabulary- and structure-wise) mobile database (17 speakers, 3 hours, 4000 utterances, 20000 words).

Table 1. Serbian speech database breakdown

	Audiole books	Mobile phone recordings
total duration	154h	61h
speech duration	129h	42h
# speakers	60+	351
# utterances	87428	74137
# words	1314574	355396
# characters	6275495	1600390
# words per utterance	15.04	4.79

5. Language model. Language modeling is a very important aspect of speech recognition systems, especially on large vocabularies. For the purpose of the experiments in this article, a trigram language model was trained on the training part of the database transcriptions — which have over 1.5 million words in total by themselves (around 121000 different word forms), as well as on an additional part coming from the Serbian journalistic corpus for more realistic estimation of probabilities (this part consists of over 440000 additional sentences, mostly from newspaper articles and similar sources, for a 40%-60% mix). The journalistic corpus was only used to provide better estimates of *n*-gram probabilities, with no new words coming from it, so in the end there were still around 121000 different words (unigrams) in the final language model, with 1.3 million bigrams and 358000 trigrams. The Kneser-Ney smoothing method [16] with a pruning value of 10^{-7} was applied to obtain the previously mentioned numbers, as it was proven to be optimal [3]. The language model was trained using the SRILM toolkit [17]. The vocabulary included words from both train and test sets (there are no out-of-vocabulary words). However, the test set was not included in the language model training procedure, to simulate real situations where the user says something not entirely expected by the speech recognizer. Test data perplexity was calculated to be 768.8.

6. Experimental setup. Several training parameters for the proposed training procedure have been examined in order to find the optimal configuration:

– number of hidden layers (7-9);

– number of neurons per layer (512-1024);

– number of training epochs and iterations (3-5 epochs);

– layer splicing options (how many *-1,0,1* vs. *-3,0,3* layers);

– HMM structure complexity (3000 vs. 4000 states);

– extending database using speed-perturbed data (Boolean);

– applying random volume adjustments to data (Boolean);

– using accent-specific vowel models (Boolean);

– using pitch as an additional feature (Boolean).

Other DNN parameters were the same everywhere. These included the basic features — 40 high-resolution Mel-frequency cepstral coefficients (MFCCs), calculated using 40 filter banks, on 30ms frames, with 10ms shifts, the number of initial and final parallel jobs (3 and 16), as well as the initial and final learning rates (0.001 and 0.0001).

Input alignments to the DNN training stage were provided via the baseline HMM-GMM speaker-adaptive training system (section 2, Figure 1), producing WER of around 22% on the described test set (without any additional discriminative training at this stage).

7. Experimental results. Initial training parameters were the following: 7 neural network layers, 625 neurons per each layer, 4 epochs (60 iterations) of training, two initial *-1,0,1* layer splicings and four *-3,0,3* splicings for the remaining hidden layers, no accent-specific vowel models, 3000-state HMM structure, no artificial database extension or adjustment, without the additional pitch features.

Various number of neurons per each hidden layer were examined for the fixed initial number of layers (7). All the produced word error rates were slightly under 10%, but the experiments have shown that 512 seems to be too few, and 1024 too many. In between, 625 and 768 (the midway between 512 and 1024) neurons per layer seem to produce very similar WERs — 9.71% and 9.72% to be exact (Figure 2). In this situation, it was reasonable to choose the less complex system — more neurons take a lot more time to train and the resulting models take up a significantly larger space on the disk, which can further result in much slower decoding time (this can be crucial for some ASR applications, e.g. on devices with limited memory resources and not so powerful processors, such as mobile phones), that can severely affect user experience when real time communication is to be expected. Specifically, total training times were between 11 hours for 512 neurons per layer, and 18 hours for 1024 neurons per layer, using a minibatch size of 32 to successfully complete the whole training on the concrete GPU. The difference in the final model size was substantial, as it ranged from 29MB (least complex system), up to almost 100MB (most complex system). The decoding time varied as well, even though it was very fast on the given machine in general, as it took only about an hour (or slightly more) to decode the whole 18 hours of test data. Compared to real time, the decoding speed ranged from 5.4% to 8.2% of real time. This is likely much more prominent on less powerful devices.

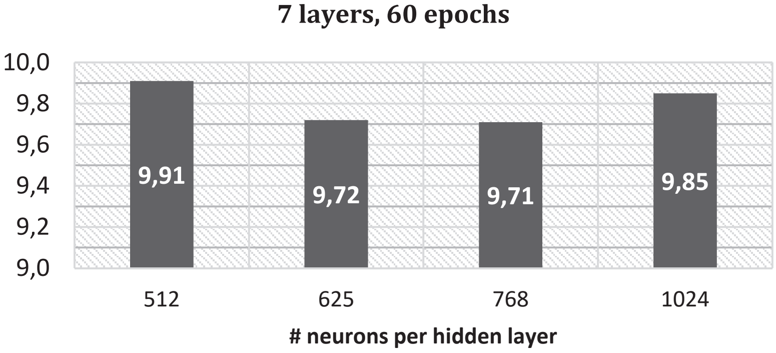

Fig. 2. WER results for different number of neurons per layer (7-layer DNN)

Character error rate mimicked the way WER behaved — it ranged from 2.53% to 2.62%. Based on this, it can be assumed that most errors occur in just a few letters (characters) of words, while most of the word (basic form, i.e., the lemma) is often correctly recognized. This is an observation that will be explained in more detail later in the article. Presenting the most frequent mistakes (particularly substitution errors) the decoder made proves the abovementioned considerations (Table 2). The most prominent insertions and deletions contain a great majority of very short one-syllable words (1-3 characters long), such as prepositions, conjunctions, etc. There are also substitutions, such as changing one version of a word pronunciation with another allowed pronunciation (which is often just slightly acoustically different). These type of errors can be handled in the pronunciation dictionary (lexicon) used for training and testing in future iterations. On the other hand, the mentioned deletions and insertions can probably only be solved with a more sophisticated language model.

Increasing the number of hidden layers in the network by one improved WER by a significant amount — to 9.45%. Further increments didn't seem to produce more improvement (Figure 3) — WER actually started going in the wrong direction. Other combinations of layers and neurons did not seem to be more successful either (8 layers and 768 neurons per layer, 9 layers and 625 or 768 neurons per layer, etc. — all fell short). So the 8-layer 625-neuron configuration seemed to be optimal one for the given amount of data and variability. Here, the training was completed in 14 hours, and the final models occupied 45MB. The decoding speed was 6.3% of real time. The character error rate also got better, from 2.53% to 2.47%. Most frequent errors expectedly remained almost the same, but the number of different errors was slightly reduced.

Table 2. Most frequent word error examples (total words in test set: 158653)

	substitutions	insertions	deletions
total	10889	952	3578
examples	je → i (1.18%)	i (7.98%)	je (17.08%)
	i → je (0.41%)	u (6.09%)	i (16.1%)
	bilo → bila (0.29%)	je (5.88%)	u (5.14%)
	peter → petar (0.26%)	na (4.41%)	a (2.91%)
	u → o (0.26%)	da (3.15%)	da (1.98%)
	koja → koje (0.25%)	mu (2.94%)	na (1.68%)
	osamnaeste → osamneste (0.25%)	ni (2.42%)	o (1.57%)
	iz → i (0.24%)	ne (1.79%)	se (1.43%)
	sam → osam (0.21%)	to (1.79%)	on (1.34%)
	me → mi (0.2%)	od (1.58%)	ona (1.09%)
	je → nije (0.19%)	a (1.47%)	s (0.87%)
	hiljadu → hijadu (0.17%)	sa (1.47%)	su (0.81%)
	je → koje (0.17%)	s (1.16%)	joj (0.75%)
	koje → koji (0.17%)	se (1.16%)	mu (0.67%)
	sa → se (0.16%)	pre (1.05%)	pa (0.67%)
	revolucija → revolucije (0.15%)	po (0.84%)	bi (0.61%)

The number of training epochs seemed to be optimal right from the start (Figure 4) — changing the number of training epochs in any direction seemed to increase WER. More epochs and iterations are definitely not suggested. If time was more relevant, the slightly shorter training (less iterations) could be proposed. The system might benefit from a change in learning rate or the number of parallel jobs (initial and/or final) alongside the change in epochs and training iterations, but it was not tested in this round of experiments, since the given learning rate was recommended for systems of similar complexity, and the training and validation probabilities did not show signs of overtraining.

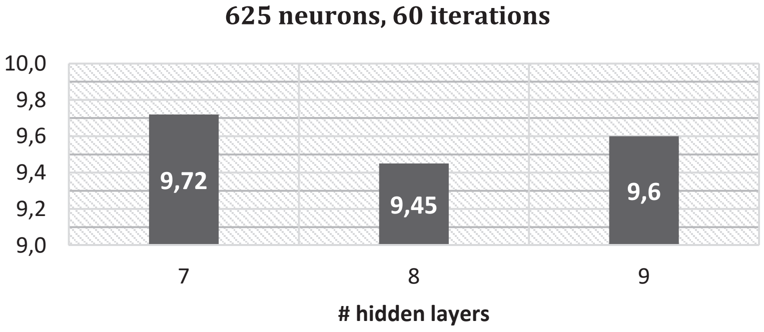

Fig. 3. WER results for different number of layers (625 neurons per layer)

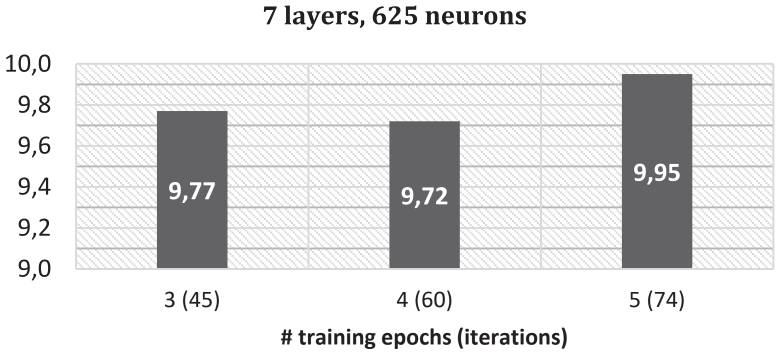

Fig. 4. WER results for different training lengths (number of iterations)

Initially, in the 8-layer neural network architecture, the two layer splicing variants were equally distributed among hidden layers (3 initial layers had the *-1,0,1* splicing with neighboring layers, while the 4 deepest layers had the *-3,0,3* splicing put in place, or *3+4* for short). Additional adjustments did not produce better results. In the 9-layer system however, an alternative *4+4* splice distribution managed to improve WER in relation to the original *3+5* splicing variant, but it still did not reach the current best result of 9.45% WER. All of these results can be seen in Figure 5.

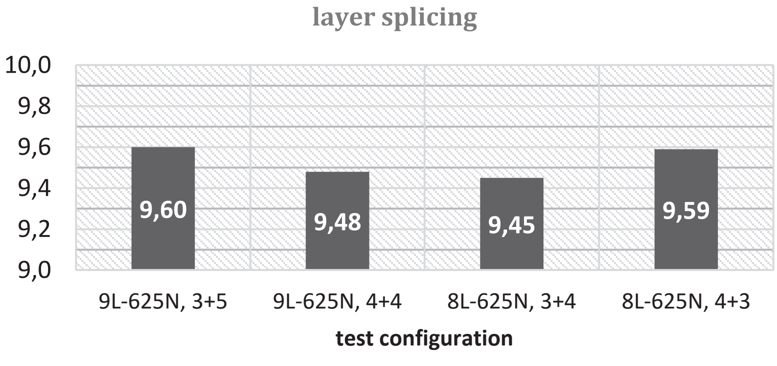

Fig. 5. WER results for different layer splicing

In all further experiments (unless explicitly mentioned), the best architecture so far was used (8 layers, 625 neurons per layer, *3+4* splicing, 4 epochs, 9.45% WER).

The next adjustment to be examined was artificial data expansion. The speech data was expanded by using speed perturbance coefficients to produce new versions of the database (on the fly), that contained either faster or slower speech than the original database. These perturbed database versions were added to the original in the training set. Several perturbance coefficients have been examined, always in pair (one slower and one faster database version plus the original). Unfortunately, no gain was made in WER, at least with the given architecture (Figure 6). The more the perturbed data was changed in comparison to the original, the worse the results became. Features were probably too dispersed in this setup for the previously optimized network to cover properly. An increase in the number of layers and/or neurons could be tested (with caution to not cause overfitting). Nevertheless, as stated before, smaller neural networks are preferred, so those experiments were skipped for now. Not to mention that the training was much longer (over 42 hours long), with a lot more iterations (same number of epochs) due to the increased amount of data.

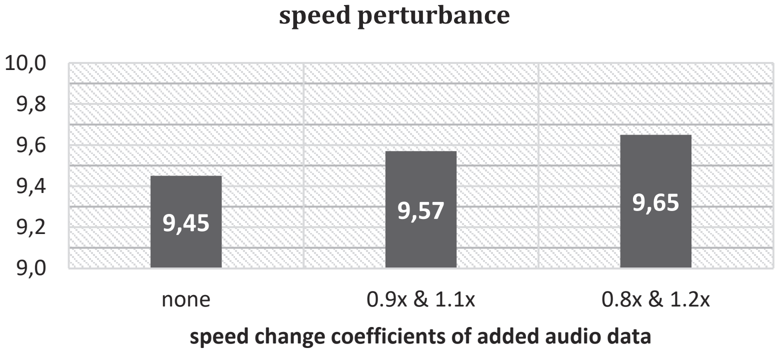

Fig. 6. WER results with added speed-perturbed data

Another interesting approach was to modify input audio data by volume scaling using a random coefficient for each file in the speech database. This was also performed on the fly in the training scripts. Therefore, there is no extra training material, only the variability is increased. This could compensate the fact that some speakers naturally speak louder than others, so in a way this may produce a more equalized database volume-wise, i.e., with more training data for much louder or much quieter speech than normal (neutral loudness). More precisely, the volume adjustment coefficient was randomly selected for each file between the values of 0.125 and 2.0. At first, this approach did not improve the best result — in fact, it produced one of the worst ones so far. Luckily, increasing the number of neurons per layer to 768 made a big difference — almost reaching the best result. Because of that result, even a 9-layer architecture was examined, and the new best result so far was obtained — 9.32% WER. All the mentioned results are presented in Figure 7.

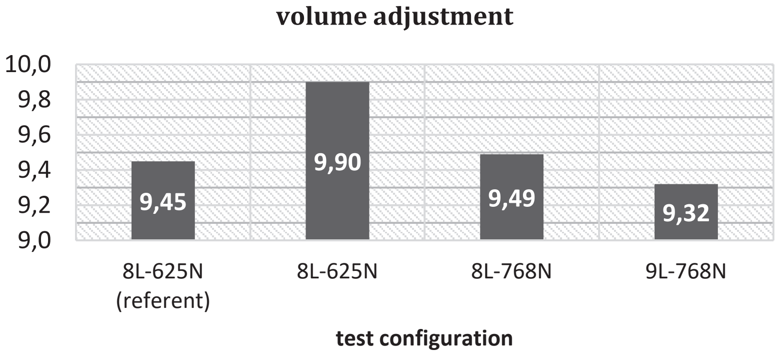

Fig. 7. WER results with volume-adjusted data

Unfortunately, this had a bit of a toll on training length and decoding speed — 18.5 hours of training, 6.5% of real time decoding speed. Yet, it has been shown that this kind of an approach can work well, if not for anything else than for systems running on more powerful machines which can perform in real time without issues. Of course, due to randomness, the training is not exactly repeatable (unless producing the same pseudorandom sequence of volume scaling coefficients using the same seed), and sometimes you can be a bit luckier than at other times. For the rest of the results in the paper, the 9.45% WER result is still considered as the referent one.

A major change incorporated in the system was obtained using separate models for vowels with different accents — five standard Serbian accents, plus the unstressed version of the vowel, ignoring diphthongs. The biggest change in this approach was creating a lexicon with word pronunciations with accents, which was eventually performed using the most comprehensive existing accentuation dictionary for Serbian. Other appropriate changes were made to tree-based clustering of HMM states as well. Of course, firstly the HMM-GMM models had to be retrained with the described changes to produce new phoneme-level alignments for the neural network. After everything has been performed, there was still no luck in improving the existing WER. Increments in layers and neurons also produced no improvement (Figure 8). Increasing the number of HMM states from 3000 to 4000 in the initial system to possibly cluster the now more diverse set of phoneme states in a better way also didn't change much, producing a WER of 9.68%. This was likely because speakers often pronounced words in unexpected, incorrect ways, so that accented vowel models were not trained in the best possible way — it can be said that there was a lot of “noise” in the data. Unfortunately, manually labeling the whole training set (specifically vowel accents) is not possible in practice, especially for datasets this big or even larger. Maybe this approach can work better on smaller databases, or where speakers were instructed to speak exclusively in the linguistically correct way (and data was checked thoroughly). Hopefully though, adding pitch features in the mix can help, since pitch and accents are very correlated in the Serbian language.

Adding the fundamental frequency alone as an additional feature helped a lot (Figure 9). Specifically, there were 3 new features — weighted log-pitch, delta-log-pitch and the warped Normalized Cross Correlation Function (NCCF) value (originally between -1 and 1, higher for voiced frames). The whole system was retrained from the start for the new feature set. Finally, a significant jump in WER was obtained, to 9.18%. At this point, the training procedure was moved to a machine with a better GPU, but for later comparisons let's mention that the training lasted around 10 hours. The final model was still around 45-46MB big, and the decoding speed was 5.1% of real time. Character error rate followed the WER improvement, and jumped from 2.47% to 2.40%. The list of the most frequent errors was still unchanged, although their frequency decreased, as the acoustic model actually helped distinguish similar words a little bit better.

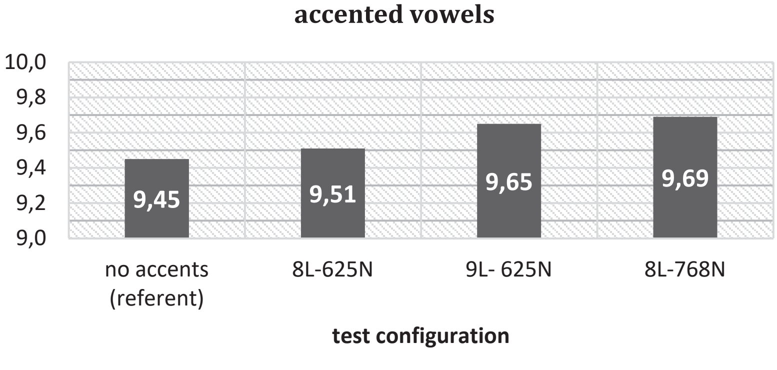

Fig. 8. WER results with accented vowel models

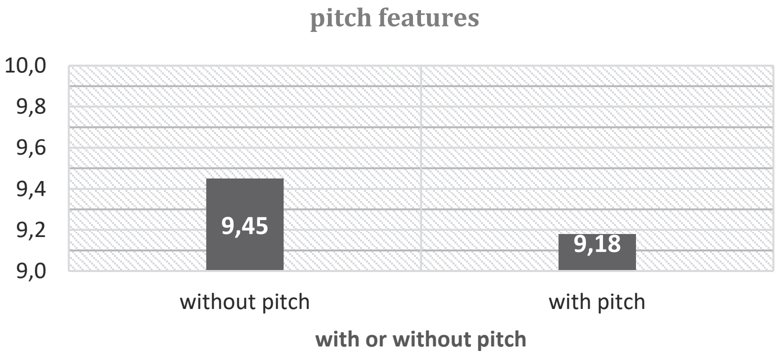

Fig. 9. WER results with added pitch features

The final TDNN-based experiment was the examination of pitch features in combination with accented vowels, for the reasons stated above. Fortunately, this produced the best result to date (Figure 10) — word error rate of 9.06% and CER of 2.37%. The whole DNN part of the training took 9.5 hours and produced a model of 45MB (just slightly smaller than without accents). The decoding capabilities were calculated to be 5.6% of real time.

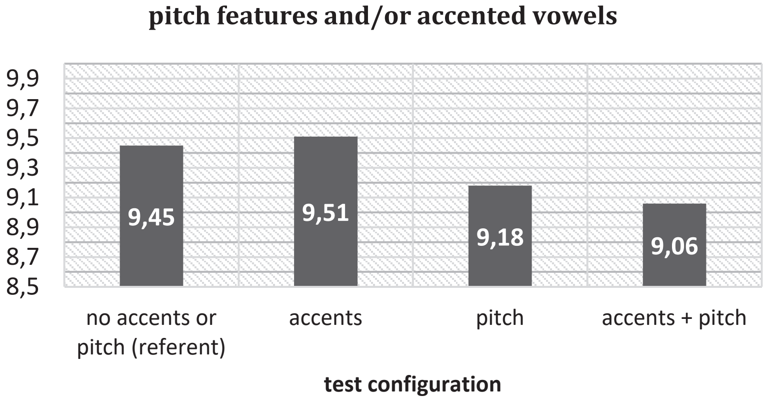

Fig. 10. WER results with added pitch features and/or accents

In order to create even better acoustic models for the given database and a general purpose LVCSR system, speaker audio data equalization could be analyzed. Purposefully adding noise to speech data to prepare models for not-so-perfect environments (this could also produce more variability to the features) might be another way. However, creating a sophisticated language model based on recurrent neural networks seems to be the most promising direction.

Let us also mention that WER was mostly accumulated on the audio books test dataset (10.21% WER, 2.63% CER), while the mobile database test produces a much better WER of less than 1% (0.82% WER, 0.33% CER for pitch-based models with accents).

Like mentioned before, overall CER is only at around 2.4%, which is due to the high language inflectivity and most likely suboptimally trained language models. High inflectivity means that small changes in words are used to express different grammatical categories, e.g. case, tense, gender, number. This creates a possibility of very similar but completely different words (with the same basic word form, i.e., lemma) to be substituted with each other in the recognition process. There is another proposed ASR system evaluation method that was created for languages like these — inflectional WER (IWER), which assigns a weight between 0 and 1 to so-called "weak" substitutions, where the lemma of the word is correctly recognized [18]. If the default weight of 0.5 is taken here, for the best system the IWER value is calculated to be 7.23%. A lot of small errors still persist — including alternative pronunciations of same words (e.g. numbers) and some errors that are more due to the language model in use, but these are not further explored in this paper.

The system also performed a lot better on the female speaker test dataset than on its male counterpart — WER of 5.66% compared to 11.22%, but this was likely a consequence of speaker choice (random selection) for this particular test set (Figure 11). Some of the male speakers do have more background noise and lower quality audio in general (e.g. mumbling).

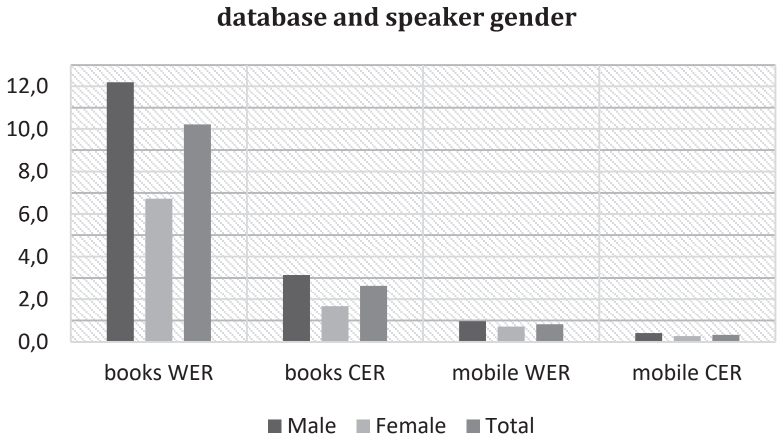

Fig. 11. WER breakdown for speech database parts and speaker genders

Figure 12 shows a side-to-side comparison between this system (the best variant that incorporates vowel accents and pitch features) with the previous ASR systems for Serbian that use acoustic models trained on the same speech database.

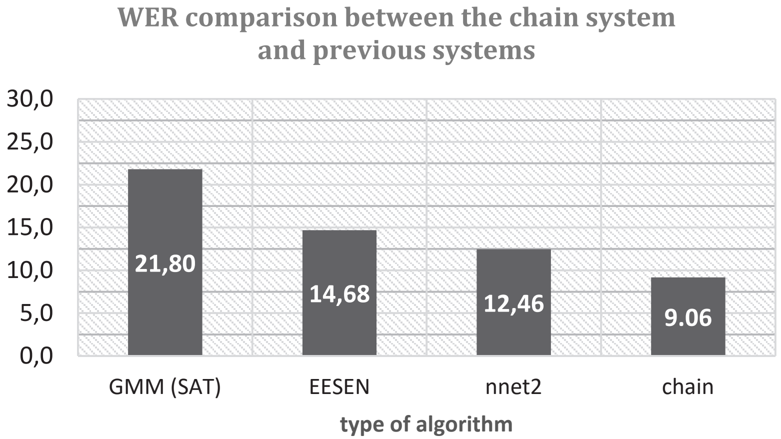

Fig. 12. WER comparison – baseline GMM, end-to-end, SGD system, chain system

As expected, the "chain" system more than halves the WER obtained with the baseline HMM-GMM system, which is described in Section 2.

The next system is an end-to-end system for Serbian developed by the Eesen framework [19]. In this system a LSTM-based deep neural network is trained to directly model connections between speech and context-independent lexicon units, which dramatically reduces the amount of expert knowledge needed to successfully train a competitive LVCSR system. The training is CTC-based, while allowing the usage of weighted finite state transducers (WFSTs) in the decoding procedure. When using the Serbian speech database from this article, as described in [20], a reasonably good WER of 14.68% is obtained. Still, the 9.06% WER of the "chain" system is superior.

The final comparison is with the "nnet2" system, which is the system based on modified SGD and parameter averaging, as detailed in [3]. The acoustic models here were also TDNNs, with such an architecture that more efficiently models longer temporal contexts [21]. This system produced another significant improvement, lowering the WER to 12.46%. It was shown in [3] that WER can improve even further by introducing a discriminative MMI training stage before the DNN phase to produce even better input alignments for the neural network, but for fair comparison the

12.46% WER was chosen. This is the second best system, right after “chain”, but it is still almost 30% worse in relative word error rate.

It can be concluded that the new system is a lot better than any of the previous ones. Not only in WER, but in speed (it uses frame subsampling) as well, as well as efficient training.

8. Experiments with LSTM. After the TDNN experiments have been completed and catalogued, several experiments using long short term memory recurrent neural network architectures have been carried out. These tests used deep LSTMs with a recurrent projection layer — unlike the regular LSTM architecture, in which there are recurrent connections in LSTM layers from cell output units to cell input units, input gates, output gates and forget gates, here, another separate linear projection layer exists after a LSTM layer, with recurrent connections attaching this new layer to the input of the LSTM layer, as detailed in [22]. This architecture is often abbreviated as a LSTMP neural network. Several versions of the neural network with different complexities were examined (number of layers and neurons), while keeping the splicing method the same throughout the experiments — the proposed *-2:2* splicing for the initial hidden layer, after which there are no spliced inputs for any of the other layers, up to, and including the output layer.

Experiments were first concentrated on less complex architectures, similar to tests in [22], but without much success (Figure 13). Only two layers didn’t seem to be enough, and tweaking the recurrent vs. non-recurrent projection dimensionality didn't produce any change at all. Also, having more layers, but remaining with a very low number of neurons per hidden layer produced the worst result by far (this configuration was tried without any projections as well).

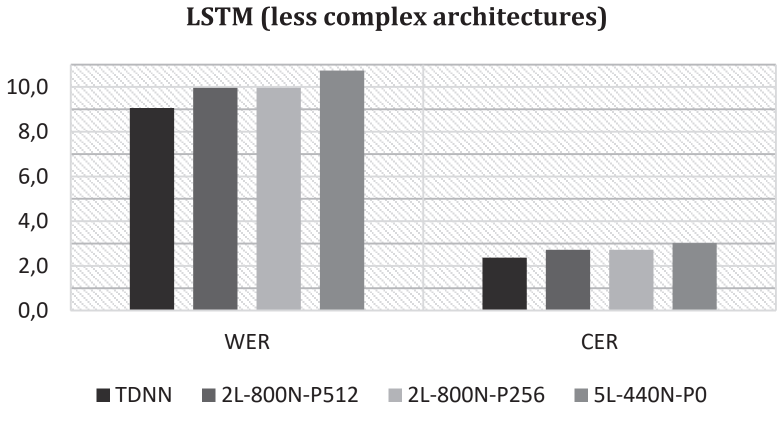

Fig. 13. LSTM results with less complex architectures

Going back to more hidden layers with a reasonable number of neurons, the results got a bit better (Figure 14). All in all, the more complex the network got, the better the results were, especially related to the number of neurons per layer. By far the best result was obtained from the 3-layer 2048-neuron system (more layers with such a big number of neurons were not examined due to very long training times). On the whole test dataset, it produced a WER of 9.00%, which is even better than the best TDNN system, with a gain on audio books, and a slight loss on the mobile dataset. Unfortunately, these architectures are very slow, compared to TDNN on the same machines, regarding both training and decoding speeds. The best LSTM architecture was being trained for almost a full week (4 epochs, 113 iterations, same learning rates as before for TDNN), and even the minibatch size had to be lowered a couple of times during training because of GPU memory errors. The final model size was 89MB, with decoding speed of 23% of real time.

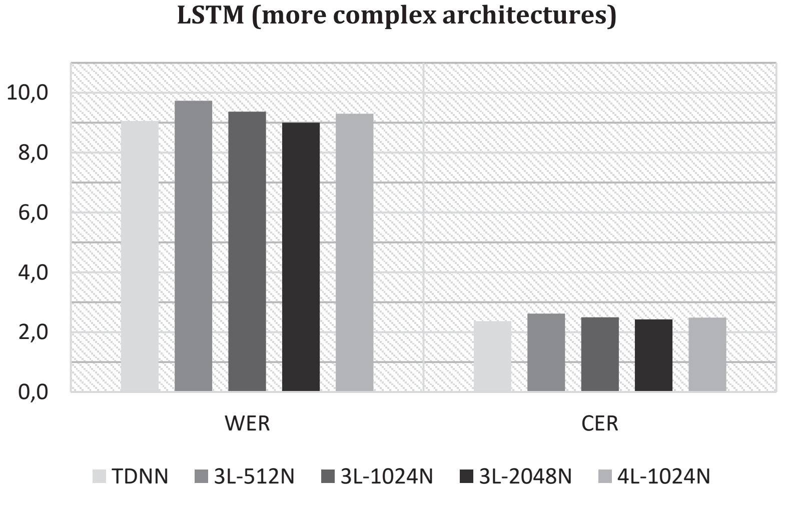

Fig. 14. LSTM results with more complex architectures

9. Conclusion. This paper describes all experiments and available results on the Serbian LVCSR speech database in detail, using the “chain” DNN models. It can be concluded that an 8-layer, 625-neuron-per-layer structure works best, without the need for artificial database expansion using speed perturbance or random volume adjustment, with the explained splicing method, while also preventing the system from overtraining. Accented vowel models in combination with additional pitch features prevailed as the best configuration until now, while even pitch features alone produced a significant improvement. Various experiments have been proposed to further polish the acoustic models. Nevertheless, finding the optimal language model configuration (also based on neural networks) and incorporating it in the final system seems to be the correct way to proceed.

References

1. Popović B., Pakoci E., Ostrogonac S., Pekar D. Large vocabulary continuous speech recognition for Serbian using the Kaldi toolkit. Proceedings of 10th Conference on Digital Speech and Image Processing (DOGS’2014). 2014. pp. 31–34.
2. Popović B. et al. Deep neural network based continuous speech recognition for Serbian using the Kaldi toolkit. Proceedings of 17th International Conference on Speech and Computing (SPECOM’2015). 2015. LNCS 9319. pp. 186–192.
3. Pakoci E., Popović B., Pekar D. Language model optimization for a deep neural network based speech recognition system for Serbian. Proceedings of 19th International Conference on Speech and Computing (SPECOM’2017). 2017. LNAI 10458. pp. 483–492.
4. Povey D., Kuo H-K.J., Soltau H. Fast speaker adaptive training for speech recognition. Proceedings of 9th Annual Conference of the International Speech Communication Association (INTERSPEECH’2008). 2008. pp. 1245–1248.
5. Povey D., Zhang X., Khudanpur S. Parallel training of DNNs with natural gradient and parameter averaging. Proceedings of 3rd International Conference on Learning Representations Workshop (ICLR’2015). 2015. arXiv:1410.7455. 28 p.
6. Graves A., Fernández S., Gomez F., Schmidhuber J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. Proceedings of 23rd International Conference on Machine Learning (ACM’2006). 2006. pp. 369–376.
7. Povey D. et al. Purely sequence-trained neural networks for ASR based on lattice-free MMI. Proceedings of 17th Annual Conference of the International Speech Communication Association (INTERSPEECH’2016). 2016. pp. 2751–2755.
8. Sak H., Senior A., Rao K., Beaufays F. Fast and accurate recurrent neural network acoustic models for speech recognition. Proceedings of 16th Annual Conference of the International Speech Communication Association (INTERSPEECH’2015). 2015. pp. 1468–1472.
9. Povey D. Discriminative Training for Large Vocabulary Speech Recognition. Ph.D. thesis. Engineering Department. Cambridge University. 2003. 170 p.
10. Sak H. et al. Learning acoustic frame labeling for speech recognition with recurrent neural networks. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2015). 2015. pp. 4280–4284.
11. Povey D. et al. The Kaldi speech recognition toolkit. Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU’2011). 2011. pp. 1–4.
12. Senior A. et al. Acoustic modelling with CD-CTC-SMBR LSTM RNNs. Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU’2015). 2015. pp. 604–609.
13. Povey D. et al. Boosted MMI for model and feature-space discriminative training. Proceedings of 33rd International Conference on Acoustics, Speech and Signal Processing (ICASSP’2008). 2008. pp. 4057–4060.
14. Su H., Li G, Yu D., Seide F. Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2013). 2013. pp. 6664–6668.
15. Suzić S., Ostrogonac S., Pakoci E., Bojanić M. Building a Speech Repository for a Serbian LVCSR System. *Telfor Journal*. 2014. vol. 6. no. 2. pp. 109–114.
16. Kneser R., Ney H. Improved backing-off for M-gram language modeling. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’1995). 1995. pp. 181–184.

17. Stolcke A., Zheng J., Wang W., Abrash V. SRILM at sixteen: Update and outlook. Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'2011). 2011. vol. 5. 5 p.
18. Bhanuprasad K., Svenson D. Errgrams - a way to improving ASR for highly inflected Dravidian languages. Proceedings of 3rd International Joint Conference on Natural Language Processing (IJCNLP'2008). 2008. pp. 805–810.
19. Miao Y., Gowayyed M., Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'2015). 2015. pp. 167–174.
20. Popović B., Pakoci E., Pekar D. End-to-end large vocabulary speech recognition for the Serbian language. Proceedings of 19th International Conference on Speech and Computing (SPECOM'2017). 2017. LNAI 10458. pp. 343–352.
21. Peddinti V., Povey D., Khudanpur S. A time delay neural network architecture for efficient modeling of long temporal contexts. Proceedings of 16th Annual Conference of the International Speech Communication Association (INTERSPEECH'2015). 2015. pp. 2–6.
22. Sak H., Senior A.W., Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Proceedings of 16th Annual Conference of the International Speech Communication Association (INTERSPEECH'2015). 2015. pp. 338−342.

Pakoci Edvin — research assistant of of the Department for Power, Electronic and Telecommunications Engineering of the Faculty of Technical Sciences, University of Novi Sad. Research interests: human-computer interaction, speech recognition and synthesis, speaker identification, numerical simulations, statistical analysis, artificial intelligence. The number of publications — 32. edvin.pakoci@uns.ac.rs; 6, Trg Dositeja Obradovića, 21000, Novi Sad, Serbia; office phone: +381214852521.

Popović Branislav — Ph.D., Dr. Sci., research associate of the Department for Power, Electronic and Telecommunications Engineering of the Faculty of Technical Sciences, University of Novi Sad, member, the Centre for Vibro-Acoustic Systems and Signal Processing (CEVAS) of the Faculty of Technical Sciences, University of Novi Sad, associate professor of Academy of Arts Belgrade, Alfa BK University, founder and owner, Computer Programming Agency Code85. Research interests: human-computer interaction, speech recognition and synthesis, speaker identification, emotion recognition, image processing, pattern recognition, clustering algorithms, numerical simulations, statistical analysis, applied mathematics, artificial intelligence. The number of publications — 60. branislav.popovic.gm@gmail.com, http://www.branislavpopovic.com; 6, Trg Dositeja Obradovića, 21000, Novi Sad, Serbia; office phone: +381214852521.

Pekar Darko Jovan — research assistant of of the Department for Power, Electronic and Telecommunications Engineering of the Faculty of Technical Sciences, University of Novi Sad, CEO (Chief Executive Officer), AlfaNum Speech Technologies. Research interests: human-computer interaction, speech recognition and synthesis, speaker identification, emotion recognition, speech morphing, numerical simulations, artificial intelligence. The number of publications — 100. darko.pekar@alfanum.co.rs; 40, Bulevar Vojvode Stepe, 21000, Novi Sad, Serbia; office phone: +381-21-485-2521.

Acknowledgements. The work is supported in part by the Ministry of Education, Science and Technological Development of the Republic of Serbia, within the project "Development of Dialogue Systems for Serbian and Other South Slavic Languages", EUREKA project DANSPLAT, "A Platform for the Applications of Speech Technologies on Smartphones for the Languages of the Danube Region", ID E! 9944, and the Provincial Secretariat for Higher Education and Scientific Research, within the project "Central Audio-Library of the University of Novi Sad", No. 114-451-2570/2016-02.

УДК 004 **DOI** 10.15622/sp.58.3

Э. Пакоци, Б. Попович, Д. Пекар

УСОВЕРШЕНСТВОВАНИЕ РАСПОЗНАВАНИЯ СЕРБСКОЙ РЕЧИ С ПОМОЩЬЮ ОБУЧЕННЫХ НА ПОСЛЕДОВАТЕЛЬНОСТЯХ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ

Пакоци Э., Попович Б., Пекар Д. **Усовершенствование распознавания сербской речи с помощью обученных на последовательностях глубоких нейронных сетей.**

Аннотация. Представлены последние усовершенствования в распознавании сербской речи, достигнутые с использованием современных глубоких нейронных сетей, основанных на применении дискриминативного обучения на последовательностях для акустического моделирования. Описываются несколько вариантов новой системы распознавания слитной речи с большим словарем (LVCSR), которая основана на обучении по критерию максимальной взаимной информации (LF-MMI) без использования решетки. Параметры системы варьировались таким образом, чтобы достичь наименьших значений ошибки распознавания слов (WER) и ошибки распознавания символов (CER) при использовании самой большой существующей речевой базы данных сербского языка и наилучшей n-граммной языковой модели общего назначения. В дополнение к настройке самой нейронной сети (числа слоев, сложности, объединения элементов слоя и т.д.) для получения наилучших результатов были исследованы и другие ориентированные на конкретный язык способы оптимизации, такие как использование акценто-зависимых моделей гласных фонем и их сочетание с тональными признаками. Также была исследована настройка речевой базы данных, которая включает в себя искусственное расширение базы данных путем изменения скорости речевых высказываний и масштабирование уровня громкости для учета вариативности речи.

Результаты экспериментов показали, что 8-слойная глубокая нейронная сеть с 625 нейронами в каждом слое работает в данных условиях работает лучше других сетей без необходимости увеличения речевой базы данных или регулировки громкости. Кроме того, тональные признаки в сочетании с использованием акценто-зависимых моделей гласных обеспечивают наилучшие показатели точности во всех экспериментах.

Ключевые слова: глубокая нейронная сеть, автоматическое распознавание речи, обучение на последовательностях, LF-MMI, акценты, основной тон, сербский.

Пакоци Эдвин — младший научный сотрудник департамента энергетики, электроники и телекоммуникационного инжиниринга факультета технических наук, Нови-Садский университет. Область научных интересов: человеко-машинное взаимодействие, распознавание и синтез речи, идентификация диктора, цифровое моделирование, статистический анализ, искусственный интеллект. Число научных публикаций — 32. edvin.pakoci@uns.ac.rs; ул. Трг Доситейа Обрадовича, 6, 21000, Нови Сад, Сербия; р.т.: +381214852521.

Попович Бранислав — д-р техн. наук, научный сотрудник департамента энергетики, электроники и телекоммуникационного инжиниринга факультета технических наук, Нови-Садский университет, сотрудник центра виброакустических систем и обработки сигналов (CEVAS) факультета технических наук, Нови-Садский университет, доцент Академии искусств в Белграде, Альфа БК университет, основатель и владелец , Computer Programming Agency Code85. Область научных интересов: человеко-машинное взаимодействие, распознавание и синтез речи, идентификация диктора, распознавание эмоций, обработка изображений, распознавание образа, алгоритмы кластеризации, цифровое моделирование, статистический анализ, прикладная математика, искусственный интеллект. Число научных публикаций — 60. branislav.popovic.gm@gmail.com, http://www.branislavpopovic.com; ул. Трг Доситейа Обрадовича, 6, 21000, Нови Сад, Сербия; р.т.: +381214852521.

Пекар Дарко Йован — младший научный сотрудник департамента энергетики, электроники и телекоммуникационного инжиниринга факультета технических наук, Нови-Садский университет, главный исполнительный директор, AlfaNum Speech Technologies. Область научных интересов: человеко-машинное взаимодействие, распознавание и синтез речи, идентификация диктора, морфинг речи, статистический анализ, искусственный интеллект. Число научных публикаций — 100. darko.pekar@alfanum.co.rs; ул. Войводе Степе, 40, 21000, Нови Сад, Сербия; р.т.: +381-21-485-2521.

Поддержка исследований. Работа выполнена при частичной финансовой поддержке Министерства образования, науки и технологического развития Республики Сербия в рамках проекта «Развитие диалоговых систем для сербских и других южнославянских языков», проект EUREKA DANSPLAT, «Платформа для приложений речевых технологий на смартфонах для языков Дунайского региона», ID E! 9944 и провинциального секретариата высшего образования и научных исследований в рамках проекта «Центральная аудиобиблиотека Университета Нови-Сад», № 114-451-2570 / 2016-02.

Литература

1. *Popović B., Pakoci E., Ostrogonac S., Pekar D.* Large vocabulary continuous speech recognition for Serbian using the Kaldi toolkit // Proceedings of 10th Conference on Digital Speech and Image Processing (DOGS’2014). 2014. pp. 31–34.
2. *Popović B. et al.* Deep neural network based continuous speech recognition for Serbian using the Kaldi toolkit // Proceedings of 17th International Conference on Speech and Computing (SPECOM’2015). 2015. LNCS 9319. pp. 186–192.
3. *Pakoci E., Popović B., Pekar D.* Language model optimization for a deep neural network based speech recognition system for Serbian // Proceedings of 19th International Conference on Speech and Computing (SPECOM’2017). 2017. LNAI 10458. pp. 483–492.
4. *Povey D., Kuo H-K.J., Soltau H.* Fast speaker adaptive training for speech recognition // Proceedings of 9th Annual Conference of the International Speech Communication Association (INTERSPEECH’2008). 2008. pp. 1245–1248.
5. *Povey D., Zhang X., Khudanpur S.* Parallel training of DNNs with natural gradient and parameter averaging // Proceedings of 3rd International Conference on Learning Representations Workshop (ICLR’2015). 2015. arXiv:1410.7455. 28 p.
6. *Graves A., Fernández S., Gomez F., Schmidhuber J.* Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks // Proceedings of 23rd International Conference on Machine Learning (ACM’2006). 2006. pp. 369–376.
7. *Povey D. et al.* Purely sequence-trained neural networks for ASR based on lattice-free MMI // Proceedings of 17th Annual Conference of the International Speech Communication Association (INTERSPEECH’2016). 2016. pp. 2751–2755.
8. *Sak H., Senior A., Rao K., Beaufays F.* Fast and accurate recurrent neural network acoustic models for speech recognition // Proceedings of 16th Annual Conference of the International Speech Communication Association (INTERSPEECH’2015). 2015. pp. 1468–1472.
9. *Povey D.* Discriminative Training for Large Vocabulary Speech Recognition // Ph.D. thesis. Engineering Department. Cambridge University. 2003. 170 p.

10. *Sak H. et al.* Learning acoustic frame labeling for speech recognition with recurrent neural networks // Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2015). 2015. pp. 4280–4284.
11. *Povey D. et al.* The Kaldi speech recognition toolkit // Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU’2011). 2011. pp. 1–4.
12. *Senior A. et al.* Acoustic modelling with CD-CTC-SMBR LSTM RNNs // Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU’2015). 2015. pp. 604–609.
13. *Povey D. et al.* Boosted MMI for model and feature-space discriminative training // Proceedings of 33rd International Conference on Acoustics, Speech and Signal Processing (ICASSP’2008). 2008. pp. 4057–4060.
14. *Su H., Li G, Yu D., Seide F.* Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription // Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2013). 2013. pp. 6664–6668.
15. *Suzić S., Ostrogonac S., Pakoci E., Bojanić M.* Building a Speech Repository for a Serbian LVCSR System // Telfor Journal. 2014. vol. 6. no. 2. pp. 109–114.
16. *Kneser R., Ney H.* Improved backing-off for M-gram language modeling // Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’1995). 1995. pp. 181–184.
17. *Stolcke A., Zheng J., Wang W., Abrash V.* SRILM at sixteen: Update and outlook // Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU’2011). 2011. vol. 5. 5 p.
18. *Bhanuprasad K., Svenson D.* Errgrams - a way to improving ASR for highly inflected Dravidian languages // Proceedings of 3rd International Joint Conference on Natural Language Processing (IJCNLP’2008). 2008. pp. 805–810.
19. *Miao Y., Gowayyed M., Metze F.* EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding // Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU’2015). 2015. pp. 167–174.
20. *Popović B., Pakoci E., Pekar D.* End-to-end large vocabulary speech recognition for the Serbian language // Proceedings of 19th International Conference on Speech and Computing (SPECOM’2017). 2017. LNAI 10458. pp. 343–352.
21. *Peddinti V., Povey D., Khudanpur S.* A time delay neural network architecture for efficient modeling of long temporal contexts // Proceedings of 16th Annual Conference of the International Speech Communication Association (INTERSPEECH’2015). 2015. pp. 2–6.
22. *Sak H., Senior A.W., Beaufays F.* Long short-term memory recurrent neural network architectures for large scale acoustic modeling // Proceedings of 16th Annual Conference of the International Speech Communication Association (INTERSPEECH’2015). 2015. pp. 338−342.