# CONTINUOUS SPEECH RECOGNITION OF KAZAKH LANGUAGE

*Orken* Mamyrbayev[1], *Mussa* Turdalyuly[1], *Nurbapa* Mekebayev[2], *Kuralay* Mukhsina[2], *Alimukhan* Keylan[1], *Bagher* BabaAli[1], *Gulnaz* Nabieva[1], *Aigerim* Duisenbayeva[2], *Bekturgan* Akhmetov[1]

[1]Institut of Information and Computational Technology, Almaty, Kazakhstan
[2]Information Technology Department, al-Farabi Kazakh National University, Almaty, Kazakhstan

**Abstract.** This article describes the methods of creating a system of recognizing the continuous speech of Kazakh language. Studies on recognition of Kazakh speech in comparison with other languages began relatively recently, that is after obtaining independence of the country, and belongs to low resource languages. A large amount of data is required to create a reliable system and evaluate it accurately. A database has been created for the Kazakh language, consisting of a speech signal and corresponding transcriptions. The continuous speech has been composed of 200 speakers of different genders and ages, and the pronunciation vocabulary of the selected language. Traditional models and deep neural networks have been used to train the system. As a result, a word error rate (WER) of 30.01% has been obtained.

## 1 Introduction

The technology of speech recognition changes the way of information access, the tasks performance, and it facilitates human labour. The growth of speech applications in recent years has been surprising and is improving every day. Due to the large number and availability of language facilities automatic speech recognition (ASR) is successfully used in life, in most Western languages like English, French, Italian, etc., and Asian languages such as Chinese, Japanese, Indian, etc. And because of the shortage or inaccessibility of these resources, this technology is less common in Central Asia.

Modern ASR systems are based on the application of a statistical approach that process large volumes of speech data for the purpose of constructing an acoustic model. Such speech data, along with their spelling transcriptions, is called an acoustic corpus. According to the type of text scoring, there are two types of acoustic corpuses - cases containing well-read material and spontaneous speech.

One of the most common acoustic corpuses for the English language, which were used in many tasks of speech recognition, are: TIMIT - corpus of phonetically representative connected speech [1], by means of which phonetic studies are conducted. Switchboard – a corpus of telephone spontaneous speech [2]. TIDIGITS - a corpus of a speaker-independent sequence recognition of numbers [3]. Aurora2 is a noisy version of TIDIGITS [4].

The Kazakh language is a low-resource language with insufficient speech and text data. One of the main problems of Kazakh language is its morphological structure. Morphological structure of this language is agglutinative. Agglutinative languages are languages that have a system, where the dominant type of word change is agglutination [5]. This means, we get new words by adding suffixes and endings. The composition of agglutinative languages includes all Turkic languages (Kazakh, Kirghiz, etc.). Another problem is the lack of available acoustic data in open access. A good speech recognition system requires many hours of acoustic data. And also there is no single standard of its sounds in the form of phonemes, dyphones, tryphons, etc.

Another biggest problem for researchers in processing Kazakh speech is lack of phonetically rich databases. Recently, attempts have been made to develop audio and text corpuses for Kazakh speech studies. Researchers of the Institute of Information and Computational Technologies have formed a database that includes a Kazakh texts, read by 200 speakers, an average 10 minutes each. Besides, they are working on a project to create a large corpus of Kazakh language, for the development of morphologically and syntactically annotated textual corpuses.

## 2 Speech recognition system

Kaldi is a tool for speech recognition with open source, written in C ++, which gives it an advantage in terms of speed. It includes OpenFst for the Finite State Transformer (FST) infrastructure and support of linear algebra BLAS and LAPACK. This tool provides complete, accessible and clearly structured documentation for the development of speech recognition systems [6].

Commonly used approaches to the removal of elements, such as Mel-frequency cepstral coefficients (MFCC) and perceptual linear predictive coefficients (PLP), are supported. And also dispersion normalization,

* Corresponding author: morkenj@mail.ru

vocal track length normalization (VTLN), linear dispersion analysis (LDA), maximum likelihood linear transformation (MLLT), and others. Adaptation of the model, that is, maximum likelihood linear regression (MLLR), as well as adaptation of acoustic models, that is, limited MLLR are provided. The topology of the hidden Markov model (HMM) can be specified separately for each context-independent sound. The roots of the decision tree can be divided between sounds and individual sound states. Consequently, Kaldi was chosen among the other candidates for speech recognition tools to develop a system for recognizing continuous Kazakh speech.

The structure of the toolkit for creating automatic speech recognition systems Kaldi [7] is shown in Figure 1.

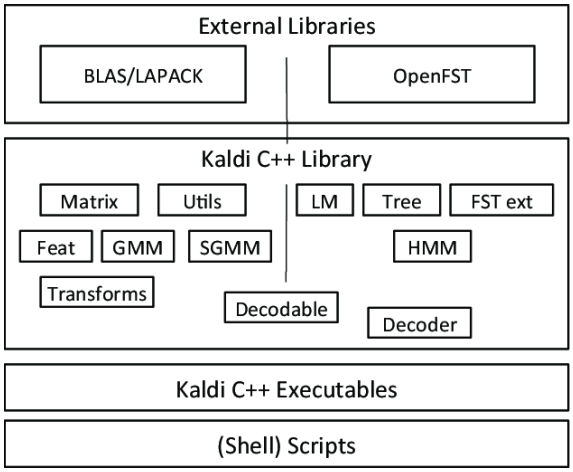


**Fig. 1.** Structure of the Kaldi ASR tool.

## 3 Speech and language processing

The overall architecture of the DNN-HMM hybrid system [9] is shown in Figure 2. DNN is trained to predict the back probabilities of each context-dependent state with these acoustic observations. When decoding the output probabilities, they are divided by the previous probability of each state forming a "pseudo-probability", which is used instead of the state probabilities in HMM [8].

The first step in training the DNN-HMM model is to prepare the GMM-HMM model using training data. The standard Kaldi receipt for acoustic modeling based on DNN consists of the following steps:

- extraction of features (13 MFCCs can be used as functions);
- training a monophonic model;
- training a triphone model with delta functions;
- training a triphone model with delta-delta-delta;
- training a triphone model with linear discriminant analysis (LDA) and maximum likelihood linear transformation (MLLT);
- adapted acoustic training (SAT), that is, training on the linear regression functional capabilities (fMLLR) with the maximum likelihood function (fMLLR);
- preparation of the final DNN-HMM model.

The DNN-HMM model is trained using fMLLR-adapted functions; The decision tree and alignment are obtained from the SAT-fMLLR GMM system. We tried DNN with two types of non-linearities (activation functions): tanh and p-norm. Generalization of the p-norm has been proposed in [8], it is calculated as follows:

$$y = \|x\|_p = (\sum_i |x_i|^p)^{1/p}, \quad (1)$$

where the vector x is a small group of inputs. The value of p is adjusted. It was shown in [8] that p = 2 gives better results. The output layer is a softmax layer with an output size equal to the number of states depending on the context (1609 in our case). DNN was trained over the FMLLR functions. The system was trained for 15 epochs with a learning frequency of 0.02 to 0.004, and then 5 epochs with a constant learning rate (0.004).

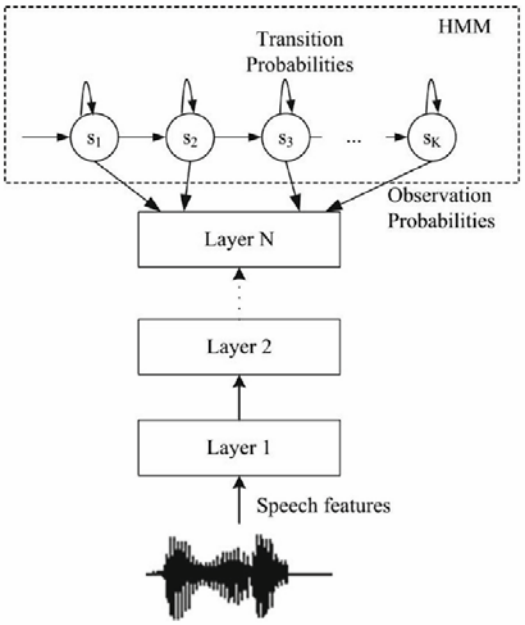


**Fig. 2.** Architecture of the DNN-HMM hybrid system.

The language model makes it possible to determine the probability frequency of word sequence. The complexity of creating a language model depends on the language. For example, the use of statistical models (N-grams) is sufficient for English languageAnd for inflectional languages (there are several types of a word), for example, the Russian language, the use of only statistical models to create a language model does not have an effect as for English languageFor reliable assessment of statistical relationships between words, a large amount of data is needed. Therefore, hybrid language models are used to model inflectional languages. They can use information about language rules, word forms and classical statistical models. The language model determines the structure of the language at the level of words.

The dictionary covers the scoring of all words in a given language model. Scoring of words are divided into

several elementary sets. For example, ұстаушы – u s t a u sh y.

In this work, the phoneme is considered as the minimum semantically distinguishing phonetic unit of a language that has no independent lexical or grammatical meaning, but serves to distinguish and identify significant units of language (morphemes and words) [10].

Vowel phonemes of the Kazakh language according to the traditional classification are: а, ә, о, ө, е, ы, і, ү, ұ. Consonant phonemes: б, г, ғ, д, ж, з, й, к, қ, л, м, н, ң, п, р, с, т, х, ш, у. When pronouncing phonemes, sometimes their variations or variations are observed. For example, in the word “kitap”, phoneme “a” is pronounced as “ə”, similarly, and “h” in the word “hatshy” - “hatchy”. However, both variants and variations do not have a meaningful function, therefore, in phonetically presented texts, variants or variations of phonemes are not considered.

**Table 1.** Font styles for a reference to a journal article.

| № | Kazakh Letters | Transliteration | № | Kazakh Letters | Transliteration |
|---|---|---|---|---|---|
| 1 | А | a | 22 | П | P |
| 2 | Ә | a | 23 | Р | R |
| 3 | Б | b | 24 | С | S |
| 4 | В | v | 25 | Т | T |
| 5 | Г | g | 26 | У | U |
| 6 | Ғ | gh | 27 | Ұ | u |
| 7 | Д | d | 28 | Ү | u |
| 8 | Е | e | 29 | Ф | Ph |
| 9 | Ё | Jo | 30 | Х | h |
| 10 | Ж | zh | 31 | Һ | h |
| 11 | З | Z | 32 | Ц | c |
| 12 | И | i | 33 | Ч | Ch |
| 13 | Й | j | 34 | Ш | Sh |
| 14 | К | K | 35 | Щ | Sh |
| 15 | Қ | Kh | 36 | Ъ | - |
| 16 | Л | L | 37 | Ы | Y |
| 17 | М | M | 38 | І | i |
| 18 | Н | N | 39 | Ь | - |
| 19 | Ң | Ng | 40 | Э | E |
| 20 | О | O | 41 | Ю | ju |
| 21 | Ө | O | 42 | Я | ja |

## 4 Data preparation

Acoustic model - creating units of acoustic models which enables the continuous-speech recognition recognition. In most cases, to determine the part of speech by sound, the phonemes of the language are selected in the form of acoustic units that affect the content of the spoken speech. The acoustic model solves the problem of converting the computed sequences of speech feature vectors into a sequence of phonetic units of spoken speech.

When creating speech recognition systems with a large vocabulary, an acoustic model is created for each acoustic unit of speech, for example, phonemes. As a rule, models consist of three states of phoneme sounding: at the beginning, in the middle and at the end. In addition, it is forbidden to switch from a late state to a previous state, but it is allowed to increase the duration of the sound on one part of the phoneme, which can remain in this state for more than one moment.

To create the system, voice data have been collected at the Institute of Information and Computational Technologies CS MES RK in Almaty. 200 people of different ages (aged 18 to 50 years) and gender were used for recording. For each speaker was prepared a text consisting of 100 sentences, which were recorded in separate files. Sentences are chosen with the richest phoneme of words in total, 36 hours of audio data were recorded. A transcription has been created - a description of each audio file in a text file. After it was created one of the important elements - the dictionary database for the speech recognition system. The dictionary consists of non-repeating words with transliteration into phonemes. The use of transliteration for Kazakh letters is given in Table 1. Audio recordings of 175 speakers were used to train the system, 16 speakers- for testing and 9 speakers- for development and customization. To create the system and perform experiments, a tool was chosen - Kaldi.

All recorded texts are collected in one file and repeated words are deleted. After they are sorted alphabetically, after- translated to phonemes. A fragment of the created dictionary is shown in Figure 3.

```
адасып   a d a s y p
адвокат   a d v o k a t
адвокаттар   a d v o k a t t a r
адзурраның   a d z u r r a n y ng
адиабаталық   a dj i a b a t a l y kh
админ   a d mj i n
адольф   a d o lj f
адресатқа   a d rj e s a t kh a
адресатынан   a d rj e s a t y n a n
адресін   a d rj e sj i n
адрестерін   a d rj e s tj e rj i n
адрестік   a d rj e s tj i k
адым   a d y m
адымын   a d y m y n
адыр   a d y r
адырға   a d y r gh a
адырдан-адырды   a d y r d a n a d y r d y
ажал   a zh a l
ажалдан   a zh a l d a n
ажалымды   a zh a l y m d y
ажалымнан   a zh a l y m n a n
ажалына   a zh a l y n a
ажар   a zh a r
ажарланды   a zh a r l a n d y
```

**Fig. 3.** Fragment of the dictionary database in Kazakh.

For Kaldi data preparation include “data" and language folders. In the data folder there is a description of our speech data, both training and testing parts are divided into two subfolders in which there are several files.

The file named "text" stores the audio file identification and their descriptions for each audio file, respectively, with the training and testing databases. And also not the auditory noises and silence were marked with the sign "! SIL".

The file "utt2spk" contains the audio file identifier and the identifiers of the corresponding speaker. In our case, due to the nature of the acoustic base, each speaker

has several utterances. Other files, such as "spk2utt", "wav.scp", etc. were built automatically using ready-made scripts.

The language folder "data / lang" is created according to this language model and contains the following files: "extra_questions.txt", "lexicon.txt", "nonsilence_phones.txt", "optional_silence.txt" and "silence_phones.txt". The language model was built on the basis of our training database, which contains 15,000 sentences. It was based on trigrams and created using the SRILM Toolkit. Kneser-Ney smoothing method without trimming was applied. Dictionary of spoken words "lexicon.txt" contains 20,000 words and their phonetic transcription, including words from both training and test sentences. The acoustic base contains about 30 hours of speech and 7 GB of memory on the disk.

The structure of the created corpus for the Kazakh language in Kaldi has the form as in Figure 4.

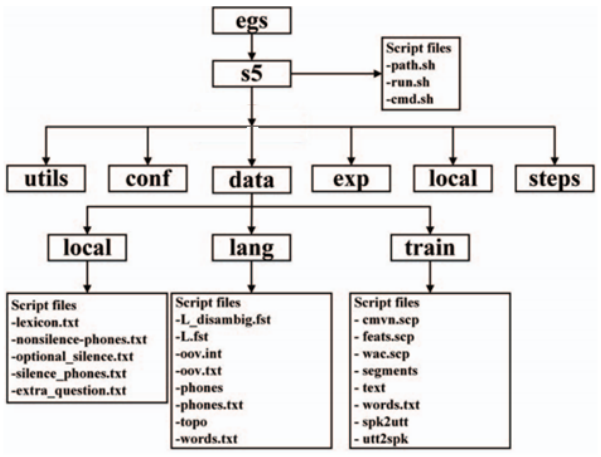


**Fig. 4.** Structure of the building for the Kazakh language in Kaldi.

## 4 Conducting experiments

In automatic speech recognition systems, the main quality indicator is the recognition accuracy, which is defined as the percentage of correctly recognized words (WRR - Word Recognition Rate) or, conversely, incorrectly recognized words (WER — Word Error Rate) [11]. Recently, the WER indicator is used as the main indicator of the accuracy of speech recognition systems, namely, its absolute value or relative if different models / systems are compared. Since with the development of speech technologies, the WER indicator is increasingly approaching zero, then the improvement of its value is more evident than increasing the accuracy of word recognition.

The method of determining the WER indicator consists in the alignment of two text lines (the first is the recognition result, and the second is a record of what was said in reality) using the dynamic programming algorithm with the calculation of the Levenshtein distance [12]. The Levenshtein distance represents the "cost" of editing the data (the minimum quantity or the weighted sum of the edit operations [13]) to convert the first line into the second with the least number of operations for manual replace (S), delete (D), and insert (I) words:

$$WER(\%) = \frac{(D+S+I)}{N} * 100(\%) \qquad (2)$$

where N is the number of words in the recognized phrase.

We started our experiment with the extraction of features, training the monophonic models using a subset of training data. The first result obtained is shown in Table 1 and denoted as "mono".

As you can see, the error in words (WER-Word Error Rate) was 62.46% for dev and accordingly, 61.36%- for test. Further for the "triphone" models, the first "tri1" using for feature extraction function MFCC and time derivatives delta and delta-delta, we obtained the following results, dev 41.61% and test 40.94%.

And the second "tri2" using LDA and MLLT, improved our results, dev 38.17% and test 38.70%. And another "tri3" to LDA and MLLT added an adaptation to SAT speakers, the result of WER improved by 1.41% for dev and 4.14% for test.

DNNs are trained using the modified Karel [4] setting on one GPU CUDA. Training is completed in two stages. During the first phase RBM is trained in a step-by-step method using the algorithm "Contrastive divergence" with 1-step selection by the Monroe-Carlo method in the Markov chain and the same functions that have already been used for the set of GMM-HMM (MFCC, energy and their first and second order derivatives).

The first RBM has Gauss-Bernoulli units and it has been trained with an initial learning rate of 0.01. Other RBM have Bernoulli-Bernoulli units and they have been trained with a training level of 0.4. Training was not controlled, the number of iterations was 3, the number of hidden layers was up to 6 and the number of units per layer up to 2048.

In the second stage, DNNs are trained using 90% of the training data for training, and the remaining 10% for evaluation. Folded RBMs from the previous phase are used to initialize DNN, and a cross-entropy criterion is used to classify individual frames into triphon states. The optimization is carried out using the standard procedure of back propagation of errors by means of a mini-batch stochastic gradient descent (SGD). To prevent reconfiguration, the target function is measured in a cross-validation set and an early stop criterion is provided. Table 2 shows the results for DNN configurations using 6 hidden layers. The optimal result of 31.78% WER was obtained for the 4 iteration

**Table 2.** Results of experiments.

| № | Model | WER (dev) % | WER (test) % |
|---|---|---|---|
| 1 | Mono (MonoPhone Training & Decoding) | 62.46 | 61.36 |
| 2 | Tri1 (Deltas + Delta-Deltas Training | 41.61 | 40.94 |

| | | | |
|---|---|---|---|
| | & Decoding) | | |
| 3 | Tri2 (LDA + MLLT Training & Decoding) | 38.17 | 38.70 |
| 4 | Tri3 (LDA + MLLT + SAT Training & Decoding) | 36.76 | 34.56 |
| 5 | DNN4_pretrain_DBN_DNN | 32.58 | 31.98 |
| 6 | DNN4_pretrain_DBN_DNN_SMBR | 33.18 | 32.06 |
| 7 | DNN4_pretrain_DBN_DNN_SMBR _i1lats_it1 | 33.60 | 32.29 |
| 8 | DNN4_pretrain_DBN_DNN_SMBR _i1lats_it2 | 33.18 | 31.88 |
| 9 | DNN4_pretrain_DBN_DNN_SMBR _i1lats_it3 | 33.32 | 32.20 |
| 10 | DNN4_pretrain_DBN_DNN_SMBR _i1lats_it4 | 32.72 | 31.78 |

## Conclusion

For the experiments, a 30-hour acoustic corpus of the Kazakh language was created. Recorded audio data of 200 speakers of different gender and ages. Experiments were conducted with known models. Different results were obtained, and the results were improved with the use of deep neural networks.

We presented experiments with DNN systems that have been trained using cross-entropy based on frames and various sequential-distinctive criteria for a 30-hour session of continuous speech. We have achieved good results in this task. The system building scripts and neural network training code are released as part of Kaldi's free open source toolkit, which allows a wider range of speech recognition researchers to use these most advanced techniques in their work.

The article has been prepared on the basis of the project: IRN AP05131207 Development of the technology of multilingual automatic speech recognition using deep neural networks.

## References

1. Garofolo, John S., et al. TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. Web Download. Philadelphia: Linguistic Data Consortium, 1993.
2. Godfrey, John, and Edward Holliman. Switchboard-1 Release 2 LDC97S62. Web Download. Philadelphia: Linguistic Data Consortium, 1993.
3. R. Gary Leonard, and George Doddington. TIDIGITS LDC93S10. Web Download. Philadelphia: Linguistic Data Consortium, 1993.
4. H.G.Hirsch, D. Pearce: â€œThe Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditionsâ, Proceedings of the ISCA workshop ASR2000, Paris, France, 2000.
5. Wikipedia. Agglutinative languages // Access mode: https://ru.wikipedia.org/wiki/Агглютинативные_языки free (accessed date 20.04.2018).
6. Access mode: http://kaldi-asr.org/doc/ free (accessed date 20.04.2018).
7. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, and J. Silovsky, "The Kaldi speech recognition toolkit," in IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (No. EPFL-CONF192584), IEEE Signal Processing Society, 2011.
8. Karpov, A., Markov, K., Kipyatkova, I., Vazhenina, D., Ronzhin, A.: Large vocabulary Russian speech recognition using syntactico-statistical language modeling. Speech Commun. 56, 213–228 (2014)
9. Kipyatkova I.S., Karpov A.A. Dnn-based acoustic modeling for Russian speech recognition using Kaldi // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) - 2016, Vol. 9811, pp. 246-253
10. Bazarbayeva Z. Fundamentals of Kazakh phonology. - Almaty: Inst. Linguistics. 2012. p - 120.
11. Wikipedia. Word error rate // Access mode: https://en.wikipedia.org/wiki/Word_error_rate free (accessed date 20.04.2018).
12. Levenshtein V. I. Binary codes capable of correcting deletions, insertions and reversals // Sov. Phys. Dokl. 1966. Vol. 6. P. 707—710.
13. Khokhlov Y., Tomashenko N. Speech recognition performance evaluation for LVCSR system // Proc. of the 14th Intern. Conf. "Speech and Computer" SPECOM—2011, Kazan, Russia. 2011. P. 129—135.