

Evaluation of Methods to Combine Different Speech Recognizers

Tomas Rasyimas
Vilnius University
Muitinès St. 8, Kaunas, Lithuania
Email: tomas.rasyimas@khf.vu.lt

Vytautas Rudžionis
Vilnius University
Muitinès St. 8, Kaunas, Lithuanian
Email: rudzionis@vukhf.lt

Abstract— The paper deals with the problem of improving speech recognition by combining outputs of several different recognizers. We are presenting our results obtained by experimenting with different classification methods which are suitable to combine outputs of different speech recognizers. Methods which were evaluated are: k-Nearest neighbors (KNN), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Logistic Regression (LR) and maximum likelihood (ML). Results showed, that highest accuracy (98.16 %) was obtained when k-Nearest neighbors method was used with 15 nearest neighbors. In this case accuracy was increased by 7.78 % compared with best single recognizer result. In our experiments we tried to combine one native (Lithuanian language) and few foreign speech recognizers: Russian, English and two German recognizers. For the adaptation of foreign language speech recognizers we used text transcribing method which is based on formal rules. Our experiments proved, that recognition accuracy improves when few speech recognizers are combined.

I. INTRODUCTION

Speech recognition applications could be subdivided into two broad classes: the applications using large vocabulary continuous speech recognition and applications using the recognition of voice commands from a predefined set of voice commands. It may seem that the first type of applications has the wider area of possible applications. But it is more complicated task to ensure the necessary recognition accuracy when using large vocabulary continuous speech recognition. At the same time there are a lot of potential applications when high accuracy of voice commands from a predefined set of allowable voice commands (may be even very big set of voice commands) is enough to achieve users satisfaction. The area of similar applications is big and such applications could be developed more rapidly than applications based on continuous speech recognition. The areas of voice commands based applications could be transport, logistic, medical and other information systems, various personal assistants, etc. It should be noted that for widely used languages (English, Spanish, German, etc.) voice recognition based applications became everyday reality and could be found in a various situations and areas. Among the well known examples we can mention set of tools distributed by Google or Nuance.

The development of large vocabulary speech recognition systems requires enormous resources: both material and human resources. It is difficult to find such resources in the countries where relatively not widely spoken languages are used as a primary mean of communication. This could be illustrated by the fact, that companies such as Microsoft,

Apple, Nuance aren't particularly interested in developing Lithuanian speech recognition systems, because Lithuanian language is not so widely used as some others and don't have significant market potential. Among the possible solutions for the problem might be to try to create own speech recognition engine, or to adapt the ones created for other languages. The proprietary recognizer has bigger potential and is more flexible solution, but this is also the more costly solution. At the same time it has been shown that proper adaptation of existing foreign language acoustic models could speed up the development of recognizer and lead to the acceptable recognition level in that language [1]–[4], [6], [7]. Some previous studies have shown that speech recognition systems of languages such as English, Spanish or Russian can be quite well adapted for Lithuanian speech recognition [1], [3], [4]. However, the recognition results are not always as good as necessary and depend on many factors. So, it is natural to try to create hybrid systems, which are based on combination of different speech recognition systems and consequently try to achieve better recognition accuracy. The essence of hybrid recognition is a parallel use of several different recognizers with the hope, that at least one of the recognizers will give the correct result and it will be possible to detect the correct answer [4]. Hybrid approach is one of the ways to achieve higher recognition accuracy in speech recognition systems. This implies combination of hypotheses provided by different recognition engines in order to get higher recognition accuracy.

The idea of creating hybrid speech recognizer and adapting other languages acoustic models is not new. These kinds of researches are especially important for all under resourced languages. There were successful attempts to estimate acoustic models for new target language using speech data from varied source languages, but only limited data from the target language [10]. Also, Google researchers show very promising results in transformation of English to other languages such as Lithuanian, French and so on. What is more, researchers are experimenting with different acoustic models adaptation methods in order to maximize the recognition performance with small amount of non-native data available [11]. Statistical algorithms for combining different acoustic models are used quite often and produces promising results [1], [3], [4], [6], [11], [12]. These researches shows, that in many cases it is possible to achieve high enough recognition accuracy by using hybrid systems with adapted acoustic models.

The paper presents our activities to adapt several foreign

language (English, German, Russian) speech recognizers for the recognition of limited Lithuanian vocabulary and evaluate some methods (k-Nearest neighbors, Linear Discriminant Analysis, Quadratic discriminant Analysis, Logistic Regression, and maximum likelihood), used for different speech recognizers combination.

Further paper is organized as follows. In Chapters I and III we are presenting method and tools used for adaptation of foreign language recognizers. In Chapter IV there is presented prototype system used in experimental evaluation experiments. Chapter V briefly summarizes the speech corpus used in recognition experiments. Finally in Chapter VI there are presented and discussed the results of experiments. In Chapter VII several conclusions are presented and discussed.

II. FOREIGN LANGUAGE RECOGNIZERS ADAPTATION

For the evaluation purposes we decided to use one native¹ (Lithuanian) and several foreign language recognizers. Among foreign language recognizers we used Russian², English³ and two German⁴ language open source speech recognizers. The adaptation procedure will be described as follows. First of all foreign speech recognizers were adapted to recognize Lithuanian commands. Adaptation was done by using formal rules method [5]. All Lithuanian commands, that were collected in this corpus, were transcribed by using foreign language phonemes. By using formal rules method a set of transcription rules were created. The structure of rules was as follows: left context; current letter; right context and list of phonetic units. This list represents foreign language sound that best matches current letter with left and right contexts. If left or right context of the rule can be any, then symbol ‘*’ was used. In this way the new written form of Lithuanian voice command was obtained. Some of the transcribing rules are listed in Table I.

TABLE I.
SOME EXAMPLES OF TRANSCRIBING RULES

Transcribing rules			
English (voxforge)	Russian	German	German (voxforge)
*;A;I;AY,AA IY	*;A;I;ay	*;A;I;ai	*;A;I;AY
*;E;I;EH IY	*;E;I;e ii	*;E;I;ei	*;E;I;EH IHH
*;O;I;OY	*;O;I;oo ii	*;O;I;oy	*;O;I;OY
*;U;I;UW IY	*;U;I;uu ii	*;U;I;ui	*;U;I;UU IHH
*;A;U;AW	*;A;U;aa uu	*;A;U;au	*;A;U;AW
*;E;U;EH W	*;E;U;ae uu	*;E;U;ee uu	*;E;U;EH UHH
*;O;U;OW	*;O;U;oo uu	*;O;U;oo uu	*;O;U;OOH UHH
*;U;O;UW AO	*;U;O;uu oo	*;U;O;uu oo	*;U;O;Y OOH
*;I;E;IY AE	*;I;E;i ae	*;I;E;ii ee	*;I;E;IHH EHH
*;I;A;IY EY	*;I;A;i ay	*;I;A;ii ai	*;I;A;IHH AY

III. METHODS USED FOR EVALUATION

We proposed a method to combine different speech

recognition engines by using neural networks algorithms [4]. Results in earlier studies showed, that this method increased speech recognition accuracy by almost 5% compared with the best results of single recognizer. As the next step we decided to evaluate other methods and to see how efficient they could be for combining different speech recognizers. We selected five methods which we think are quite good for this task: k-Nearest neighbors (KNN), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Logistic Regression (LR) and maximum likelihood (ML). These methods were selected because of their efficiency and well studied properties.

IV. HYBRID SPEECH RECOGNITION PROTOTYPE

For evaluation of the selected methods hybrid speech recognition system prototype was developed. Python programming language was used for its development. Block diagram of such system is showed in Fig. 1.

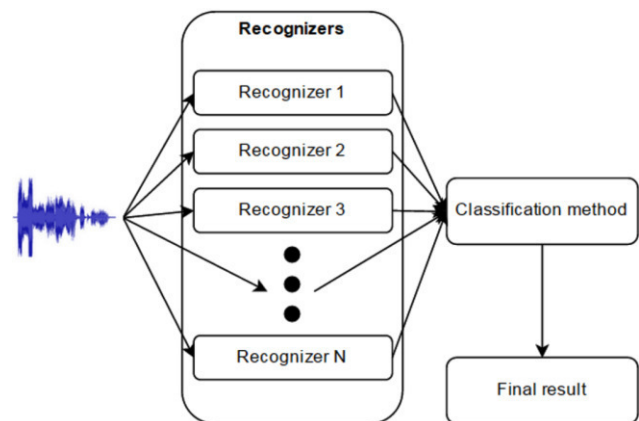


Fig. 1. Block diagram of hybrid speech recognition system.

As could be seen in the prototype, voice command is passed to all speech recognizers in parallel. After that, all recognizers produces output. Output of the recognizer is the hypothesis: score of how well audio signal matches the acoustic model [8]. This hypothesis score is passed to classification algorithm and it makes final decision.

To develop speech recognizers, PocketSphinx toolkit was used. PocketSphinx is a lightweight speech recognition engine, specifically tuned for handheld and mobile devices, though it works equally well on the desktop computers and notebooks. It is distributed under the same permissive license as Sphinx toolkit itself. Algorithmically this is hidden Markov model based speech recognition framework, which provides simple way for creating custom speech recognition systems [8].

For the quicker classification methods realization, we used scikit-learn library [9]. Scikit-learn is an open source machine learning library for the Python programming language. It realizes various classifications, regression and clustering algorithms including support vector machines, logistic regression, naive Bayes, random forests, gradient boosting, k-means and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy [9].

V. SPEECH CORPUS

A speech corpus of 25 drug names and 25 names of dis-

¹ Downloaded from https://github.com/mondhs/lt-pocketsphinx-tutorial/tree/master/impl/models/hmm/lt_cd_cont_200

² Downloaded from <http://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/Russian%20Voxforge>

³ Downloaded from <http://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/English%20Voxforge>

⁴ Downloaded from <https://www.lt.informatik.tu-darmstadt.de/de/data/open-acoustic-models> and <http://goofy.zamia.org/voxforge/de>

eases was used. Speech commands, collected in the corpus, are shown in the Table II.

TABLE II.
SPEECH CORPUS USED FOR METHODS EVALUATION

ANALGINAS	RADIREKSAS	ARTERIŲ EMBOLIJA
BIFOVALIS	RANIGASTAS	ARTERINĖ HIPERTENZIJA
CYKLODOLIS	TRACHISANAS	ARTERIŲ TROMBOZĖ
ENARENALIS	TRAVATANAS	ARTROZĖ
FERVEKSAS	TRENTALIS	ATEROSKLEROZĖ
GASTROVALIS	TRILEPTALIS	ATOPINIS DERMATITAS
HEKSORALIS	VALOKORDIN LAŠAI	BIPOLINIS AFEKTINIS SUTRIKIMAS
HEMATOGENAS	VERDINAS	BLAUZDOS KAULŲ LŪŽIAI
KETANOVAS	AIDS	BRONCHŲ ASTMA
KETONALIS	AKIŲ NUDEGIMAI	CELIULITAS
KREONAS	AKTINOMIKOZĖ	CHEMINIAI NUDEGIMAI
METFORALIS	ALERGIJA	CISTITAS
MIKARDIS	ALKOHOLIO TOKSINIS POVEIKIS	CUKRINIS DIABETAS
NEBIKARDAS	ANAFILAKSINIS ŠOKAS	DANTŲ DYGIMO SINDROMAS
PANANGINAS	ANKILOZINIS SPONDILITAS	DANTŲ DYGIMO SUTRIKIMAI
PREDUKTALIS	ANTRINĖ GLAUKOMA	DANTŲ VYSTYMOŠI SUTRIKIMAI
PROPODEZAS	APELSINO ŽIEVELĖ	

Speech corpus, used in the experiments, was gathered by recording speech of 12 people (5 female and 7 male). Each of these speakers pronounced each command name 20 times at sampling rate 16 kHz in a single session. So, every command was pronounced for 240 times. Vocabulary of all commands used in this experiment is listed in Table II.

It should be noted, that the corpus, used in these experiments, is the part of the bigger medical terms Lithuanian speech corpus. The selection of this particular set of voice commands was based on the fact, that 25 commands were those voice commands, which resulted in the highest number of recognition errors using proprietary Lithuanian speech recognizer, while the additional 25 commands were selected randomly.

VI. EXPERIMENTAL EVALUATION OF DIFFERENT SPEECH RECOGNIZERS COMBINATION METHODS

For the evaluation of methods, we used the developed prototype and described speech corpus. All acoustic models used in the recognition experiments were derived without the use of the speech corpus presented in Chapter V. So the recognition experiments were performed in speaker independent mode. Default PocketSphinx configuration was used for evaluation.

First of all, single recognizers were tested using obtained recordings. Recognition results are shown in Table III.

TABLE III.
SINGLE RECOGNIZERS ACCURACY

Recognizers	Accuracy, %
Lithuanian	89.26
Russian	81.32
English (voxforge)	88.30

Recognizers	Accuracy, %
German	81.38
German (voxforge)	90.38

Best results were obtained using German recognizer from voxforge repository. Other recognizers, such as Lithuanian and English (voxforge), showed similar recognition accuracy too. Accuracy of other recognizers was above 80 %, but lower than above mentioned recognizers.

Before the experiments, we thought that Russian recognizer will be one of the best, because Russian language and Lithuanian language have a lot similar sounds, but as results shows, our guess failed.

Later all the selected speech recognizers combination methods were trained using obtained recordings. 168 recordings were used for training and 72 recordings for testing. After training, selected methods accuracy was evaluated. The obtained results are presented in the Table IV.

TABLE IV.
ACCURACY OF COMBINED SPEECH RECOGNIZERS

Combination method	Accuracy, %
k-Nearest neighbors (11)	89.70
k-Nearest neighbors (15)	98.16
k-Nearest neighbors (21)	89.70
Linear Discriminant Analysis	93.16
Quadratic Discriminant Analysis	98.05
Logistic Regression	93.60
Maximum likelihood	89.70

Results shows, that three methods (k-Nearest neighbors (11), k-Nearest neighbors (21) and maximum likelihood) can't be used for speech recognition engine combination, because obtained accuracy is lower than best single recognizer. Other methods are suitable for speech recognizers combination. Best results (98.16 %) were acquired, when k-Nearest neighbors (15) method was used. It is very interesting, that such a simple classifier as k-Nearest neighbors generated the best results. We think that it is because of data used to evaluate selected classification methods. As we know, k-Nearest neighbors classifier requires a small amount of training data to estimate the necessary parameters. We are planning to increase number of data used for classification methods evaluation and repeat experiments to see if our guess is right. Detailed commands recognition accuracy is displayed in Table V (results were rounded to fine integer values).

TABLE V.
RECOGNITION ACCURACY % OF EVERY COMMAND

Command	k-Nearest neighbors (11)	k-Nearest neighbors (15)	k-Nearest neighbors (21)	Linear Discriminant Analysis	Quadratic Discriminant Analysis	Logistic Regression	Max hypothesis
ANALGINAS	69	86	69	78	82	79	69
BIFOVALIS	85	99	85	96	99	96	85
CYKLODOLIS	97	100	97	99	100	99	97

Command	k-Nearest neighbors (11)	k-Nearest neighbors (15)	k-Nearest neighbors (21)	Linear Discriminant Analysis	Quadratic Discriminant Analysis	Logistic Regression	Max hypothesis
ENARENALIS	100	100	100	97	100	99	100
FERVEKSAS	99	99	99	100	99	99	99
GASTROVALIS	99	99	99	99	99	99	99
HEKSORALIS	99	100	99	99	99	99	99
HEMATOGENAS	97	97	97	97	100	97	97
KETANOVAS	99	100	99	99	100	99	99
KETONALIS	92	99	92	93	96	94	92
KREONAS	82	89	82	83	92	83	82
METFORALIS	71	99	71	96	99	97	71
MIKARDIS	100	100	100	100	100	100	100
NEBIKARDAS	96	100	96	100	100	100	96
PANANGINAS	92	92	92	89	92	90	92
PREDUKTALIS	97	99	97	96	97	96	97
PROPODEZAS	97	97	97	97	97	97	97
RADIREKSAS	96	97	96	88	99	88	96
RANIGASTAS	94	99	94	97	99	97	94
TRACHISANAS	100	100	100	99	100	99	100
TRAVATANAS	100	100	100	100	100	100	100
TRENTALIS	94	96	94	93	94	93	94
TRILEPTALIS	93	96	93	94	96	96	93
VALOKORDIN LAŠAI	100	100	100	90	100	92	100
VERDINAS	97	97	97	97	97	97	97
AIDS	0	100	0	69	93	74	0
AKIŲ NUDEGIMAI	90	97	90	81	100	81	90
AKTINOMIKOŽĖ	93	100	93	99	100	99	93
ALERGIJA	74	100	74	86	100	88	74
ALKOHOLIO TOKSINIS POVEIKIS	76	99	76	92	100	92	76
ANAFILAKSINIS ŠOKAS	86	100	86	89	100	89	86
ANKILOZINIS SPONDILITAS	84	100	84	96	100	96	84
ANTRINĖ GLAUKOMA	82	99	82	78	99	78	82
APELSINO ŽIEVELĖ	90	100	90	99	100	99	90
ARTERIŲ EMBOLIJA	81	92	81	92	89	93	81
ARTERINĖ HIPERTENZIJA	92	100	92	97	100	97	92
ARTERIŲ TROMBOZĖ	93	99	93	99	100	99	93
ARTROZĖ	89	97	89	71	96	72	89
ATEROSKLEROZĖ	82	99	82	94	100	94	82
ATOPINIS DERMATITAS	92	100	92	92	100	92	92
BIPOLINIS AFEKTINIS SUTRIKIMAS	100	100	100	97	100	97	100
BLAUZDOS KAULŲ LŪŽIAI	99	99	99	85	100	86	99
BRONCHŲ ASTMA	51	96	51	89	97	90	51
CELIULITAS	97	100	97	97	100	97	97
CHEMINIAI NUDEGIMAI	100	99	100	93	99	93	100
CISTITAS	96	100	96	94	100	97	96
CUKRINIS DIABETAS	99	100	99	100	100	100	99

Command	k-Nearest neighbors (11)	k-Nearest neighbors (15)	k-Nearest neighbors (21)	Linear Discriminant Analysis	Quadratic Discriminant Analysis	Logistic Regression	Max hypothesis
DANTŲ DYGIMO SINDROMAS	99	100	99	99	100	99	99
DANTŲ DYGIMO SUTRIKIMAI	100	100	100	100	100	100	100
DANTŲ VYSTYMOŠI SUTRIKIMAI	97	97	97	97	97	97	97

We calculated average accuracy of every command and results showed, that almost 58 % of all commands are recognized with 95 – 100 % accuracy, 22 % with 90 – 95 % accuracy, 14 % with 80 – 90 % accuracy and 6 % of all commands are recognized with 40 – 80 % accuracy.

VII. CONCLUSIONS

The results of our experiments showed, that it could be reasonable to use k-Nearest neighbors (15) or Quadratic Discriminant Analysis methods to combine different speech recognizers using open source PocketSphinx based recognizers. Comparing with the best single recognizer and the best combined speech recognizers, average error was decreased by 7.78 %. In some cases, even bigger increase of recognition accuracy has been observed.

Foreign language speech recognition adaptation shows, that English, German, Russian recognizers could be quite good adapted for Lithuanian voice commands recognition.

One of the interesting areas for further research could be investigation of how different acoustic models from different language could be used to recognize the same Lithuanian voice command.

In the future, we are planning to increase recognition accuracy by finding better transcriptions to recognize Lithuanian commands using foreign languages speech engines. Also, it is necessary to increase size of the vocabulary used in the experiments. Especially important is to increase the variety of the phonetic elements used in the adaptation process.

REFERENCES

- [1] R. Maskeliūnas, A. Rudžionis, K. Ratkevičius, V. Rudžionis, "Investigation of Foreign Languages Models for Lithuanian Speech Recognition", *Electronics and Electrical Engineering*, no. 3(91), pp. 15–20, 2009.
- [2] V. Rudžionis, G. Raškinis, A. Rudžionis, K. Ratkevičius, "Comparative Analysis of Adapted Foreign Language and Native Lithuanian Speech Recognizers for Voice User Interface", *Electronics and Electrical Engineering*, vol. 19, no. 7, pp. 90–93, 2013.
- [3] V. Rudžionis, G. Raškinis, A. Rudžionis, K. Ratkevičius, G. Bartišiūtė, "Web Services Based Hybrid Recognizer of Lithuanian Voice Commands", *Electronics and Electrical Engineering*, vol. 20, no. 9, pp. 50–53, 2014.
- [4] T. Rasymas, V. Rudžionis, "Combining Multiple Foreign Language Speech Recognizers by using Neural Networks", *Human Language*

- Technologies – The Baltic Perspective*, IOS Press, doi:10.3233/978-1-61499-442-8-33, pp. 33–39, 2014.
- [5] P. Kasparaitis, “Transcribing of the Lithuanian Text Using Formal Rules”, *Informatica*, vol. 10, no. 4, pp. 367–376, 1999.
- [6] P. Kasparaitis, “Lithuanian Speech Recognition Using the English Recognizer”, *Informatica*, vol. 19, no. 4, pp. 505–516, 2008.
- [7] V. Rudžionis, K. Ratkevičius, A. Rudžionis, G. Raškinis, R. Maske-
liūnas, “Recognition of Voice Commands Using Hybrid Approach”,
ICIST2013, CCIS 403, Springer-Verlag Berlin, pp. 249–260, 2013.
- [8] D. Huggins-Daines, M. Kumar, A. Chan, A. W Block, M. Ravishan-
kar, A. I. Rudnicky, “Pocketsphinx: a free, real-time continuous
speech recognition system for hand-held devices”, *IEEE ICASSP
2006 Proceedings*, vol. 1, pp. 185–188, 2006.
- [9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion,
O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Van-
derplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot,
Duchesnay, “Scikit-learn: Machine Learning in Python”, *The Journal
of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [10] T. Schultz, A. Waibel, “Language-independent and language-adaptive
acoustic modeling for speech recognition”, *Speech Communication*
35 (1), 31–52, 2001.
- [11] Z. Wang, T. Schultz, A. Waibel, “Comparison of Acoustic Model
Adaptation Techniques on Non-Native Speech”, *IEEE International
Conference on Acoustics, Speech, and Signal Processing (ICASSP)*,
pp. 540–543, 2003.
- [12] H. Meneido, J. Neto, “Combination of acoustic models in continuous
speech recognition hybrid systems”, *Proceedings of the
International Conference in Spoken Language Processing*, vol. 9, pp.
1000–1029, 2000.