Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/978-1-61499-442-8-3

Full-duplex Speech-to-text System for Estonian

Tanel ALUMÄE [1]
Institute of Cybernetics, Tallinn University of Technology, Estonia

Abstract. The paper describes a distributed online speech-to-text system. The main features of the system are real-time speech recognition and full-duplex user experience, meaning that the partially recognized utterance is progressively displayed to the user during speaking. Other benefits include easy client-server communication protocol and system scalability to many concurrent user sessions. The paper also describes two Estonian speech-to-text applications based on the developed framework: a general-domain dictation application with an estimated word error rate of 26.4% and a radiology report dictation system with a word error rate of 13.7%. The system is open-source and based on free software.

Keywords. Speech recognition, Estonian, radiology, client-server, open source

Introduction

We have designed and implemented a distributed client-server based speech-to-text system that is used in two Estonian large vocabulary continuous speech recognition (LVCSR) applications. Its design has been driven by the current and anticipated requirements for a (possibly massively) multi-user realtime dictation system. The system relies heavily on the open-source Kaldi speech recognition toolkit [1]. The main features of the system are (close to) real-time speech recognition, full-duplex user experience (meaning that the partially recognized utterance is progressively displayed to the user during speaking), easy client-server communication protocol that can be implemented in pure Javascript and system scalability to many parallel user sessions. The system is free and open source[2].

In recent years, several automatic speech recognition (ASR) systems have been developed within the Estonian language technology national program. Both offline (not realtime) [2] as well as online (realtime) [3] speech recognition applications have been made available for free to the general public. The presented system provides several improvements over the previously developed online system. The previous system was designed for decoding relatively short speech utterances (up to 20 seconds). Although decoding was performed in an online fashion (i.e., it was done in parallel to recording the signal), the result was displayed to the user once decoding was finished (i.e, without showing the partial recognition hypotheses as feedback). The current system can de-

[1] Corresponding Author: Tanel Alumäe, E-mail: tanel.alumae@phon.ioc.ee

[2] Available at `http://github.com/alumae/kaldi-gstreamer-server`

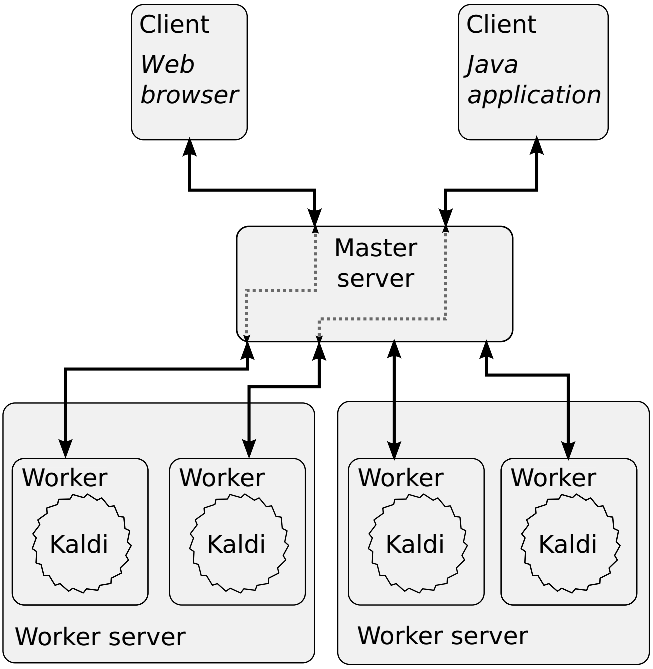

Figure 1. System architecture.

code arbitrarily long speech and provide results on-the-fly. While the previous system supported many concurrent decoding sessions, its parallelism was limited to the number of CPU cores on the machine it was deployed on. The system described in this paper supports unlimited parallelism as worker processes can be started or stopped on local or remote machines dynamically based on anticipated user activity.

The paper gives an overview of the system architecture and design principles. It also describes two speech recognition applications that are based on the system: a web application for general domain desktop dictation and an early prototype of a radiology report dictation system. We describe the creation of acoustic and language models for the two applications and present some experimental results regarding word and character error rates.

Experiments with Estonian ASR in the radiology domain have been made before [4]. A promising word error rate of less than 10% was reported. Experiments in this paper are based on an entirely new system and updated methodology. Contrary to the previous work where radiologists dictated existing written radiology reports, the evaluation in this paper is based on speech dictated spontaneously based on previously unseen radiology pictures in routine clinical practice.

1. System Overview

1.1. Architecture

The framework consists of a server component that takes care of decoding speech and normalizing the recognized hypotheses, and a client component that is responsible for recording speech and presenting the recognition results to the user.

The speech recognition server consists of a single master and multiple workers, as shown in Figure 1. The server can be concurrently accessed by multiple clients. The server components are implemented in Python, whereas the client can be implemented in any modern programming language. We provide Javascript, Java and Python based clients.

More specifically, the server component consists of two parts: a master server and a worker pool. The master server, implemented using the Tornado web framework[3], maintains a set of connected workers and their states, forwards client's audio to the workers and sends recognition results submitted by the workers back to the client. The master server also includes functionality to accept human-corrected recognition transcripts from the clients which we plan to use for adapting speech recognition models in the future.

Workers handle actual speech decoding using the GStreamer online decoder plugin of the Kaldi toolkit [1]. Workers are grouped into pools which can be dynamically started and stopped on remote servers, making it possible to handle a very high number of parallel recording sessions.

The workers can be configured to apply a user-defined command for post-processing recognition hypotheses before they are sent to the client. We are using this functionality for reconstructing compound words from recognized sub-word tokens, for converting dictated numeric expressions to numbers and times and for reconstructing some more common abbreviations from the corresponding words (such as *jne* 'etc' from *ja nii edasi* 'and others').

1.2. System Interactions

The communication protocol between the client and server is based on websockets which enable bi-directional communication over a single TCP connection. Bi-directional communication is needed for seamless full-duplex speech recognition where speech signal is sent to the server while intermediate decoding results are sent back to the client. Client can send audio in any container and encoding supported by the GStreamer framework (e.g., Ogg, MP4, Speex, etc). Recognition results are sent back to the client using the JSON encoding.

Communication between the master server and workers is also based on websockets.

The Kaldi online decoder that is employed in the workers segments incoming speech on-the-fly into utterance-like regions, separated by silence. For each segment, the decoder produces several progressively longer partial recognition hypotheses and one final hypothesis. Once a final hypothesis is generated, the decoder continues with the next speech segment. After the final hypothesis for the final speech segment is generated and sent to the client, the client-server connection is closed.

2. Applications

The speech recognition architecture is currently used in two applications: general-domain desktop dictation and radiology report dictation. The first application is a publicly available free service developed within the Estonian language technology national

[3] `http://www.tornadoweb.org/en/stable/`

program, available at `http://bark.phon.ioc.ee/dikteeri/`. The second application was implemented as part of a project dedicated to creating an Estonian speech recognition based system for dictating radiology reports.

2.1. Acoustic Models

For training the Estonian acoustic models (AMs), we used the following Estonian speech corpora, about 135 hours in total:

- the BABEL dictated speech database (8h) [5];
- a corpus of Estonian broadcast news, consisting of radio and TV news programs with both dictated speech as well as interviews (30h);
- a corpus of broadcast conversations, containing talkshows and telephone interviews from different Estonian radio stations (52h);
- a corpus of local TED-like conference speeches and university lectures, recorded with a close-talking microphone (37h);
- a corpus of studio-recorded spontaneous monologues and dialogues (7h);
- a corpus of live usage data from our Android speech recognition application *Kõnele* [3] (2h)

The AM inventory contains 43 phoneme models, a silence/noise model and a garbage model that is used to absorb unintelligible and foreign language words during training. Although different noises and fillers are annotated in our training data at a relatively fine-grained level, they are all mapped to a single silence/noise model during training. We also merge palatalized and unpalatalized versions of several phonemes into single acoustic units, since it is difficult to derive the correct palatalization from the orthographic word forms. Estonian is a quantity language: all vowels and consonants, with some exceptions, can occur in short and long segmental duration. We create distinctive models for short and long variants of all phones except */j/*. Estonian language has actually three distinctive quantity degrees: short, long and overlong [6]. However, the distinction between long and overlong duration is a property of the word foot rather than phone, and is thus difficult to model using purely segmental units. Therefore, we ignore this distinction in our acoustic models. The other reason behind such simplifications is the fact that the distinction between palatalized and unpalatalized phonemes as well as the distinction between the long and overlong quantity degree, is usually not needed for discrimination between orthographic word forms (i.e., palatalization and overlong quantity, with some exceptions, is not visible in orthography).

Acoustic model training is based on the Kaldi Switchboard training recipe. The triphone Gaussian mixture models for online decoding have 4,000 decision tree leaves and 100,000 densities. The boosted MMI objective function [7] is used for training.

2.2. Language Models

For training the language model for general domain dictation, we used around 300M words from various sources.

As Estonian is a heavily compounding and inflective language, the lexical variety of the language is very high. To reduce the out-of-vocabulary (OOV) rate of the LM, compound words are decomposed into compound segments, using a morphological analyzer

Table 1. Language model training corpora for the general domain dictation application.

Corpus	*#Tokens*	*Weight*
Newspapers	204M	0.17
Magazines	29M	< 0.01
Parliament transcripts	15M	< 0.01
Web news portals	76M	0.06
Fiction	29M	0.14
Blogs and social media	27M	0.06
Broadcast conversations	0.47M	< 0.01
Conference speeches	0.26M	< 0.01
Broadcast news	0.13M	< 0.01
Android dictation app usage data	0.006M	0.57
Total	381M	1.00

[8], and the LM is built from the decomposed tokens. After decoding, the most probable reconstruction of compound words is found using a hidden event N-gram LM.

A vocabulary of 200K units is used, selected from the mixture of corpora using maximum likelihood combination based on the development text. From each corpus, a 4-gram LM is built using modified Kneser-Ney smoothing. The individual LMs are interpolated into one by using interpolation weights optimized on development data. Finally, the LM is heavily pruned to less than one tenth in size using entropy pruning. The sizes of the text corpora (after splitting compound words) and the optimized interpolation weights in the final LM are listed in Table 1.

The radiology domain language model was trained on about 10M words, originating from written radiology reports that were produced within a year in one hospital. The texts were normalized using hand-written rules converted to character-level weighted finite state transducers (WFSTs). The normalization process included converting numbers, dates, times to words and expanding abbreviations (such as *cm*, *ml/h*) to words. Since many expansions are ambiguous (e.g., *ml* can be expanded to different inflected forms such as *millimeeter*, *millimeetri*, *millimeetrit*, depending on the context), the expansion WFST was composed with a character level pruned 20-gram LM trained on 1000 manually normalized radiology reports. This was done using the Thrax toolkit [9]. From the normalized texts, all words occurring more than once were selected as the LM vocabulary, resulting in a lexicon of 52K units. A 4-gram LM was built using modified Kneser-Ney smoothing. Unlike for the general domain dictation system, no compound splitting was performed for the radiology system.

2.3. Pronunciation Dictionary

As Estonian is a language with a close relationship between word orthography and pronunciation, a rule based system is used for deriving the pronunciations for words in the LM lexicon [4]. The rules are mainly concerned with determining the correct variant (short or (over)long) of the plosives based on the usage context. For many common foreign proper names and abbreviations, pronunciation is created by first transforming the lexical form to a localized form using a transliteration table, and then applying the common pronunciation rules.

[4] Available at `http://github.com/alumae/et-g2p`

Table 2. Word error and character error rates for the two applications.

Application	WER(%)	CER(%)
General-domain dictation	26.4	11.2
Radiology dictation	13.7	5.7

3. Evaluation

The quality of the general domain dictation system was evaluated on 215 recordings, originating from real usage data of the previously developed Android speech recognition application [3]. At the time of writing the article, we didn't have real usage data of the new web-based dictation application itself and the Android speech recognition application data was used as a close proxy. This test data consists of largely very short (one or two word) utterances. Recognition quality of longer utterances is actually expected to be better as for long utterances there is more data available to perform online cepstral mean normalization robustly.

For the radiology dictation system, the test data was collected from 18 professional radiologists in real clinical environment. Before the test, the initial version of the radiology dictation system was developed and deployed. Then, the radiologists were asked to use the system to produce reports for previously unseen studies. An instruction manual, describing how to use the prototype and how to dictate different text components, was given to test users. Speech recognition was done in real time during dictation. Every report was checked by the radiologist immediately after dictation and incorrectly recognized words or phrases were corrected. Both the speech signal and the corrected texts were stored on the speech recognition server. The corrected texts were later manually rechecked and used together with the corresponding audio signal as test data. This resulted in a test set of 363 reports, more than nine hours in total. The reference transcripts contained 24595 words in total. The initial ASR system was then evaluated and analyzed for more frequent errors. The error analysis revealed some systematic problems in the ASR system, mostly concerning the way some non-standard words (such as dates, abbreviations and acronyms) were normalized for the language model. Such errors were fixed and the collected audio was recognized again in simulation mode. This process was repeated for several iterations.

The final word error rate (WER) and character error rate (CER) of the two dictation systems based on the described test sets is listed in Table 2.

Figure 2 shows the average WER calculated over radiology reports from different modalities. The results are in concordance with earlier studies that have shown the non-radiography modalities, including magnetic resonance imaging and angiographic examinations, to have a higher risk of recognition error [10]. According to the feedback from the radiologists, the ASR system could be taken into usage as a daily tool, which enables shortened reporting and turnaround times, assuming that recognition quality of ASR will be improved.

4. Conclusions and Future Work

We have developed a scalable web-based speech recognition architecture, based on the Kaldi toolkit. The server component consists of a master and one or more worker pools.

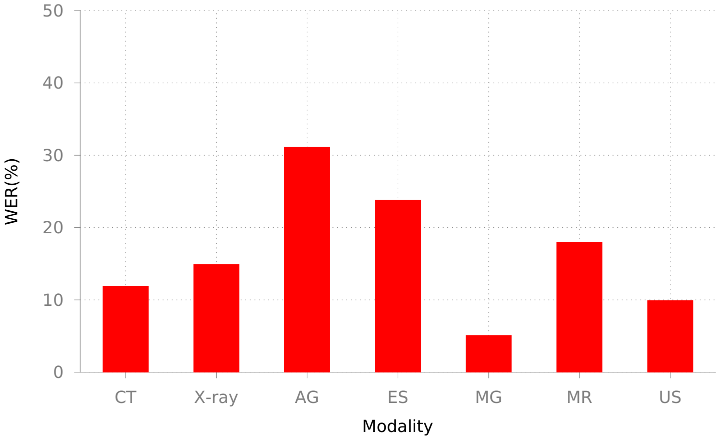

Figure 2. Average WER over radiology reports from different modalities: computed tomography (CT), X-ray, angiography (AG), endoscopy (ES), mammography (MG), magnetic resonance tomography (MR), ultrasound (US).

The master server delegates recognition tasks from clients to workers that can be deployed on one or more remote computers. System client is very lightweight and can be implemented in Javascript and executed in a modern web browser.

The system is currently used in two applications. The general-domain dictation system is available for general public at `http://bark.phon.ioc.ee/dikteeri/`. System word error rate is evaluated to be around 26%. Another system was developed for the radiology domain, having word error rate of around 14%.

Although the reported word error rates are promising, both systems need to be significantly refined to be useful in daily usage. Both dictation systems should benefit from in-domain acoustic and language model training data. The acoustic models are currently largely based on broadcast and lecture speech. We have found that when using a dictation system, people tend to use much lower voice intensity. Also, speech tempo during dictation varies from very fast to very slow. We plan to collect real usage data of both applications and use it to adapt the acoustic and language models. Both systems would also benefit from adapting the models to individual users. We are also planning to start using deep neural network (DNN) based acoustic models instead of GMMs that should result in a noticable drop in word error rate.

Acknowledgements

This research was supported by the European Union through the European Regional Development Fund and by the Estonian Ministry of Education and Research target-financed research theme No. 0140007s12.

References

[1] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *IEEE ASRU Workshop*, Dec. 2011.

[2] T. Alumäe, "Recent improvements in Estonian LVCSR," in *SLTU 2014*, (Saint Petersburg, Russia), 2014.

[3] T. Alumäe and K. Kaljurand, "Open and extendable speech recognition application architecture for mobile environments," in *SLTU 2012*, (Cape Town, South Africa), 2012.

[4] T. Alumäe and E. Meister, "Estonian large vocabulary speech recognition system for radiology," in *Baltic HLT*, (Riga, Latvia), pp. 33–38, 2010.

[5] A. Eek and E. Meister, "Estonian speech in the BABEL multi-language database: Phonetic-phonological problems revealed in the text corpus," in *Proceedings of LP'98*, vol. 2, (Prague, Czech Rep.), pp. 529–546, 1999.

[6] P. Lippus, *The acoustic features and perception of the Estonian quantity system.* PhD thesis, Tartu University, 2011.

[7] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *ICASSP 2008*, pp. 4057–4060, 2008.

[8] H.-J. Kaalep and T. Vaino, "Complete morphological analysis in the linguist's toolbox," in *Congressus Nonus Internationalis Fenno-Ugristarum Pars V*, (Tartu, Estonia), pp. 9–16, 2001.

[9] B. Roark, R. Sproat, C. Allauzen, M. Riley, J. Sorensen, and T. Tai, "The OpenGrm open-source finite-state grammar software libraries," in *ACL 2012 System Demonstrations*, (Jeju Island, Korea), pp. 61–66, July 2012.

[10] C. A. Chang, R. Strahan, and D. Jolley, "Non-clinical errors using voice recognition dictation software for radiology reports: A retrospective audit.," *Journal of Digital Imaging*, no. 24, pp. 724–728, 2011.