Cross-Lingual Adaptation of Broadcast Transcription System to Polish Language Using Public Data Sources

Jan Nouza, Petr Cerva, Radek Safarik

Technical University of Liberec
Studentska 2, Liberec, Czech Republic
{jan.nouza, petr.cerva, radek.safarik}@tul.cz

Abstract

We present methods and procedures designed for cost-efficient adaptation of an existing speech recognition system to Polish. The system (originally built for Czech language) is adapted using common texts and speech recordings accessible from Polish web-pages. The most critical part, an acoustic model (AM) for Polish, is built in several steps, which include: a) an initial bootstrapping phase that utilizes existing Czech AM, b) a lightly-supervised iterative scheme for automatic collection and annotation of Polish speech data, and finally c) acquisition of a large amount of broadcast data in an unsupervised way. The developed system has been evaluated in the task of automatic content monitoring of major Polish TV and Radio stations. Its transcription accuracy (measured on a set of four complete TV news shows with total duration of 105 minutes) reaches almost 80 %. For clean studio speech, its accuracy gets over 92 %.

Keywords: speech recognition of Polish, broadcast monitoring, acoustic model training, cross-lingual adaptation

1. Introduction

Within the last 15 years we have been working on the development of a robust automatic speech recognition (ASR) system for Czech. Its recent version is capable of fairly accurate real-time speech recognition even if the lexicon size exceeds 500,000 words. It has been used in several applications, e.g. voice dictation programs, broadcast monitoring systems, or automatic transcription of a huge historical audio archive (Nouza et al., 2014). It has been a natural idea to utilize the existing modules and acquired experience to port the system to other Slavic languages.

A few years ago we began to work on Slovak language, which is the most similar one. A prototype was presented in 2008 when it achieved 75 % word recognition rate (WRR) on a Slovak broadcast news task. The most recent version operates with WRR value around 86 % and it has been already deployed in several practical applications.

The next language we decided to focus on has been Polish. It is partly understandable to Czech people though it has completely different lexicon and phonology. We have developed a set of procedures that allowed us to utilize the existing text and audio processing tools and even the Czech acoustic model to build a Polish ASR system within a relatively short period of one year. We have saved much human labor by automating the most tedious works, such as speech data collection, phonetic annotation and acoustic model training. Moreover, during the development we have used only data (texts and recordings) that are freely available on Internet, which also reduced the costs. In this paper, we present our approach and methods in more detail.

2. State-of-the-art and related work

With its approx. 40 million native speakers, Polish is the second largest Slavic language (after Russian). Yet, there is not much literature concerned with Polish speech recognition. Major scientific databases offer a rather small number of research papers published on that topic, including those dealing with small vocabulary tasks, such as (Ziółko et al., 2011) or (Koržinek and Brocki, 2007) . A large-vocabulary continuous speech recognition (LVCSR) system for Polish is presented in (Marasek, 2003). The author used the open-source HTK toolkit to build an experimental system with a 20k-word lexicon and tested it on read speech recordings provided by 12 speakers with average WRR around 87 %. Another LVCSR system, named Skrybot, is briefly described in (Pawlaczyk and Bosky, 2009). Its decoder is based on open-source Julius system and the authors state that its WRR was 73 % in an 5-hour test. (No information about the lexicon size and the test is provided.) A more recent approach to the development of a Polish dictation system for legal texts is described in (Demenko et al., 2012).

Polish ASR has been investigated also by research teams from abroad. It is one of the 20 languages whose spoken data have been collected within project called Globalphone (Schultz, 2002). These data were later used to test a method for rapid development of language models (LM) in 5 Slavic languages, including Polish (Vu et al, 2010a). However, the most critical task in the development of any ASR system is the creation of an acoustic model (AM). Löof et al. (2009) proposed a method for cross-lingual adaptation and unsupervised iterative training of a Polish AM. Their work was part of a project focused on automatic transcription of EU parliament talks. They used recordings of Polish representatives and interpreters together with official text documents to adapt an ASR system originally designed for Spanish to Polish. With a 60k word lexicon they were able to get close to 82 % WRR on that given task.

Our approach described in this paper has a similar idea. We want to utilize the existing Czech ASR as a starting point from which the target Polish system is built in an iterative and almost fully automated way. Our goal is more ambitious because we want to create a system for transcription of TV and radio programs, where many speakers, various speaking styles and different topics can occur. The lexicon must be much larger (at least 250k words) and also the AM and LM need to be more flexible and robust.

3. Modular ASR system

The LVCSR system we have built for Czech has a modular structure where the language specific modules (lexicon, LM and AM) are separated from the rest of the system (a signal processing front-end and a decoder).

The signal parameterization unit can accept many major audio formats (e.g. WAV or MP3), which are internally converted into a stream of 16 kHz, 16 bit sampled data. They are converted into 39 Mel-frequency cepstral coefficients (MFCC) and then floating-window Cepstral Mean Subtraction (CMS) is applied. The acoustic model uses triphone multi-gaussian HMMs to represent all phonemes and 8 types of noise. The most recent version employs also a deep neural network (DNN) with 5 to 7 hidden layers which proved to be more robust especially for lower-quality signals. The decoder is based on highly optimized implementation of Viterbi algorithms. On recent CPUs, most applications can run in real time.

When porting the system to Polish, we need to replace the Czech lexicon, AM and LM by the Polish ones.

4. Lexicon and Language model for Polish

The first step consists in the acquisition of a sufficiently large text corpus, which is necessary for creating a representative lexicon and an LM.

4.1. Text corpus

Nowadays, the best source of multi-domain texts are web-pages of major newspapers and broadcasters. We have developed a web parser that can be easily adjusted to any web source type. It is based on an SGML parser that transfers an HTML file into an XML structure, from which we can distill the content we are interested in. In this way we have collected and processed almost 3 GB of texts from major Polish newspapers (*Gazeta Wyborcza, Rzeczpospolita, Dziennik Gazeta Prawna, Fakt*, etc) and TV/radio stations oriented on news (*TVP, TVN24, TV-Nowa, Polsat*).

The downloaded data were cleaned and pre-processed. The remaining HTML artifacts as well as non-literal symbols, strings, formatting marks, etc., were removed. Next we tried to replace digits by their text equivalents. This is a challenging tasks in all Slavic languages because a digit (or a string of digits), when spoken, can get various morphological forms that depend on long context. We were able to convert correctly only some types of digit strings, namely dates and years.

4.2. Lexicon

As Polish is a highly inflected language with many word-forms derived from a lemma, we had to limit the lexicon to the most frequent words. We selected all that were seen in the corpus at least 10 times and got a lexicon with 303k entries.

4.3. Pronunciation

Polish, similarly to other Slavic languages, has a rather straightforward relation between orthography and pronunciation. We used the basic rules mentioned by Demenko et al. (2003) to make a grapheme-to-phoneme (G2P) converter. It was applied to all items to get a pronunciation vocabulary needed for an ASR system. A special care was put to abbreviations (where we used a spelled-letter converter), terms with digits (e.g. 'A1') and to loanwords. For the latter, we borrowed their pronunciation from the Czech lexicon. Some words were assigned multiple phonetic variants, e.g. in case of 'NHL', 'ABC' or 'Jacka', where Polish as well as English pronunciation can occur. The recent version of the lexicon has 303,321 entries with 318,888 pronunciations. We use a set of 36 phonemes, each represented by a single-letter symbol - for illustration see Fig. 1.

4.4. Language model

The LM is probabilistic, based on N-grams. From practical reasons (mainly with respect to the very large vocabulary size), we prefer bigrams. In the 3 GB corpus of Polish texts we found 65 million different word-pairs.

5. AM building - Methods and Procedures

Building a robust AM requires that at least 50 hours of speech from hundreds of speakers must be collected. Each recording need to be annotated on the acoustic-phonetic level (as a sequence of phonemes and noises). This is the most tedious and time consuming part of the development. There exist projects that focus on the data collection, e.g. (Demenko et al., 2008). However, these data are not freely available and we had to search for alternative resources. The most suitable ones seem to be archives of broadcast stations or national parliaments. They contain both audio and text documents that can be used as source data for automatically annotated speech. For this purpose we created several procedures and schemes that are described in the following text.

5.1. Basic procedures

The procedures employ an existing ASR system (i.e. a system available at that phase) to do most works that would be otherwise done by a human expert. They cut long audio documents (at proper instants) to get shorter and manageable files, transcribe them on the orthographic and phonetic level and decide which files could be automatically added to a training set and which should be possibly corrected by a human annotator.

5.1.1. Automatic transcription of audio signal

Here we employ the basic operation mode of the ASR decoder. It takes a parameterized signal, decodes it (using the given lexicon, AM and LM) and translates it into text. Our decoder can reveal also detailed acoustic-phonetic transcription of the signal with pronunciation of each recognized word, and detected noises. This type of the output can be utilized for annotations that are necessary for AM training. Moreover, the decoder can provide also start/end times (so called time stamps) for each word and noise. We call the procedure `DoTanscription`.

5.1.2. Automatic segmentation of audio signal

The above mentioned time stamps are useful if we work with long audio documents and we need to split them into shorter segments that are better suited for further processing. It is done by procedure `DoSegmentation` that reads the detailed ASR output and cuts the signal at convenient instants - usually during silence, noise or breath, so that the speech itself is not disrupted. We can set the limits for the segment length.

Fig.1 - Program to check transcribed sentences. One can easily compare ASR output, reference text and ASR produced phonetic transcription (including silence denoted as '-' and noises indicated by digits). Differences are highlighted. In this example, the first difference is due to wrongly typed word 'ana' (error in reference), the second was made by the ASR system (omitted 'a').

5.1.3. Segmentation matched to text

This is the most essential procedure. It is used if we have an audio signal and a text that more or less corresponds to the content of the signal. In the optimal case, the text is verbatim transcription, but it can be just a brief summary. In any case, we want to find those parts of the signal that match (as well as possible) the provided text. These are searched by aligning the ASR output to the text via an algorithm proposed by Nouza et al (2013). The found segments are cut off (as in 5.1.2) and stored together with the matched text fragments in a `StackList`. The match score is computed via eq. (1). At this stage, we do not insist on perfect (100 %) match, as the segments will pass repeated decoding with a gradually improving AM later. Instead, we keep all the segments whose score is higher than a threshold (usually 70-80 %). The procedure is called `DoMatchedSegmentation`.

5.1.4. Automatic check and optional correction

This procedure takes the matched segments and classifies them into 2 sets: In the first, there are the segments that achieved 100 % score. Their phonetic transcriptions are considered correct and hence they are moved to the AM training list (`TrainList`). The other are ordered according to their scores and prepared for optional manual inspection. This is the only instant where a human may (but does not need to) enter the automated process.

In order to minimize human work we have developed a program whose interface is shown in Fig. 1. It utilizes the *ordered list* of imperfectly matched segments, and *shows and plays* them to the annotator. The words where the ASR output and the reference text differ are highlighted. The annotator just decides which is correct and clicks on it to fix the error. When needed, he/she types the correct word or modifies the pronunciation. If a segment contains speech which is not clear, it can be skipped or definitely removed from the list. The correction process is easy and fast. Moreover, it does not require a person who knows the language. Within an hour it is possible to check and correct several hundreds of speech segments, because most contain just 1 or 2 errors. The corrected segments are automatically added to the `TrainList`. The other remain in the `StackList`. In our schemes we name this procedure `CheckAndCorrect`.

5.1.5. Acoustic model retraining

When the number of newly acquired (and annotated) segments in `TrainList` is sufficiently large, we add them to the previously collected speech data and run a procedure that retrains the AM using the standard HMM training tool. We denote this step as `Retrain`.

5.1.6. Switching between phoneme sets

As one of our schemes utilizes the cross-lingual part, we need auxiliary procedures that make switching between two phoneme sets, one of the source language (SL) and another for the target one (TL). Usually, they are applied at the beginning and end of the bootstrapping phase. In the former case, we need to map all the phonemes from the TL (Polish in this case) to those of the SL with an existing AM (e.g. Czech). This approximation is only temporal and it is not much critical. We use the phoneme map proposed in (Nouza and Bohac, 2011). After its application, we get the Polish lexicon represented by Czech phonemes.

When the bootstrapping phase is finished, we switch back to the original lexicon. All the phonetic annotations made within the phase are changed to the original Polish phonetic set, using the lexicon as a reverse look-up table.

The two procedures are denoted as `MapPhonemes` and `RemapPhonemes`.

5.2. Data annotation and AM training schemes

Here we present 3 schemes, each suited for a specific use.

5.2.1. Iterative data annotation and AM training

This scheme is applied in a situation when we have a large number of speech documents and each of them is associated with some text. The goal is to find the speech segments that match parts of the text, annotate them and use them for AM retraining. The scheme combines the basic procedures in an iterative loop. We suppose that at the start we already have an AM for the target language. At the end of each iteration, new annotated data are added to the training list and a new (better) AM is trained. With this AM we repeat the scheme either from the start (step 1, i.e. a new segmentation) or for the already segmented files (step 2). The former is useful when the initial AM was trained on a small amount of data. The scheme is finished when the number of newly annotated segments is too small to run another iteration.

```
IterativeRetraining:
1 For each Document
    DoMatchedSegmentation
2 For each Segment from StackList
    DoTranscription
    CheckAndCorrect
3 Retrain
4 Repeat from step 1 or 2
```

5.2.2. Cross-lingual iterative training

This scheme is a modification of the previous one. It is used for initial bootstrapping when no AM for the target language is available. In this case, we utilize an AM from a SL and do the temporary phoneme mapping. Moreover, we need to add a certain amount of training data (e.g. 10 hours) from the SL to the `TrainList` to ensure proper performance of the HMM training procedure. After that the standard iterative scheme is started. When finished, all the annotations are remapped, the SL data are removed from the `TrainList`, and the target language AM is retrained once again.

```
CrosslingualTraining:
1 MapPhonemes
2 Add SL Data to Trainlist
3 IterativeRetraining
4 RemapPhonemes
5 Remove SL Data from Trainlist
6 Retrain
```

5.2.3. Unsupervised data acquisition and training

This scheme is used when only audio data are available. In this case, the previous two schemes cannot be applied because they have no text to be matched. Here we utilize an idea which is similar to that proposed by Vu et al, (2010b). A segment is transcribed by several different recognizers and if all the transcriptions are same, we consider them as correct and add these segments (with their annotations) to the training list. In fact, all the used recognizers have the same structure, but they have different AMs (trained on different data subsets), or differently set operating parameters. Usually, this scheme is used when we already have a mature AM for the target language. Though, it can be used also with recognizers that operate with AMs borrowed from different languages as shown in the above mentioned paper. If we want to be sure about the reliability of the scheme, we can check the transcriptions using the tool from Fig.1.

```
UnsupervisedTraining:
1 For each Document
    DoSegmentation
2 For each segment
3   For each recognizer
      DoTranscription
    If all ASR outputs same
      Add to TrainList
4 Retrain
```

6. Practical implementation and evaluation

When building a robust AM for Polish we combined all the three schemes as described further.

6.1. Bootstrapping

For the initial phase we used the large archive of Polish Sejm, namely the video files and stenograms available at *http://www.sejm.gov.pl*. The video files contain speech of good quality provided by hundreds of speakers. The stenograms are almost verbatim transcriptions of the talks, yet sometimes slightly smoothed or reformulated. (E.g. without repeated words or phrases, with synonyms, etc). They also contain some non-verbal information e.g. about reactions from the auditorium. Anyway the amount of provided data is huge.

We have chosen 20 random sessions from period 2013-2014. The stenograms were converted to plain text files and added to the corpus. Some frequent OOV names and specific words were inserted to the lexicon and the LM was recomputed. After that we started the cross-lingual training scheme described in section 5.2.2. To initialize the process, we used the Czech AM and put 10 hours of annotated Czech speech to the train list. After the first iteration, we got 1450 segments with an average duration 3.5 s, i.e. 84 minutes of annotated Polish speech data. A new (Czech-Polish mixed AM) was trained and used for improved resegmentation. In several subsequent loops we gained around 2000 new segments per iteration, from which about one third passed the manual check and correction. The scheme was stopped when the number of newly acquired segments dropped below 100. At that time 16.127 segments (17.9 hours) were available. They were used to train the first genuine Polish AM.

6.2. Standard iterative retraining process

As the next step, we took other 10 Sejm sessions and used them in the standard (mono-lingual) scheme as in section 5.2.1. In this way, we acquired an additional amount of 15.6 hours of speech. It would be possible to get much more, however we did not want to saturate the AM by one type of data. Instead, we searched for another source. We found several radio programs that have both audio and text (approximate transcription) on their web, e.g. *http://www.polskieradio.pl/Rozmowy-Jedynki*. We processed them in the same way and got 8.2 hours.

6.3. Unsupervised retraining process

To increase the variety of the training data, we had to search for other sources with a large amount of speech. Since our target application domain is broadcasting, we focused on major TV stations and their news programs. Unfortunately, these have no accompanying text and we had to use the unsupervised scheme proposed in 5.2.3. We employed 4 recognizers, each trained on a different subset of the training data. The scheme processed some 120 news programs (each about 30 minutes long) from major Polish TV stations and eventually produced 16.4 hours of annotated data. We checked manually a subset of them and found that the transcriptions (when approved by the 4 different recognizers) were fully correct for 9 of 10 segments. In most cases, the errors were marginal with a minimal impact on the trained AM. At the end of this phase we received an AM trained on 58 hours of speech.

6.4. Evaluation

To evaluate the quality of the Polish ASR system and to document the progress after each phase, we have prepared a large test set. It is made of 4 news shows from Polish major TV stations (*TVN-Fakty, TVP2-Panorama, TVP1-Wiadomosci and Polsat-Wydarzenia*). The shows are complete; from the opening jingles to the closing ones. They include all types of speech occurring in news programs: clean speech read in studio, speech with background music or noise, spontaneous utterances recorded in streets, or a dubbed speech with a talk in a foreign language in background). A Polish native speaker has made their verbatim transcriptions that were used as

references. Because we wanted to learn how the system performs under ideal conditions, we extracted a smaller subset which contained only the clean speech from studio. The main parameters of the two sets are listed in Table 1. (Term OOV denotes Out-of-vocabulary.)

	Full shows	Studio only
Total duration [min]	105	23
Number of words	14,742	3,984
OOV words [%]	0.92	1.03

Table 1. Test sets (4 full TV shows and their extracts)

During each development phase we run tests to evaluate the progress of AM training. We measured transcription accuracy using the standard formula for word recog. rate

$$WRR = (H - I)/N \cdot 100 \quad (1)$$

where N, H and I are numbers of words in the reference text, hits and insertions, respectively.

The most relevant results are summarized in Table 2, where the amount of automatically collected training data and WRR values for the two test sets are listed. The first row shows the initial situation where no Polish data were available, in the second one there are the results after the bootstrapping phase (section 6.1), next are those after the phase described in 6.2. When the last phase (6.3) finished, we got 58.1 hours and trained two AMs: one based on standard gaussian HMM and the other on deep neural networks. The DNN performs significantly better, particularly for noisy and low quality parts of the TV shows, as stated also in (Seps et al., 2014).

		WRR [%]	
Acoustic model	Hours	Full	Studio
Czech (before bootstrapping)	-	50.4	63.2
Polish after bootstrapping	17.9	61.2	78.3
Polish after further retraining	41.7	68.7	84.6
Polish final GMM model	58.1	74.1	91.3
Polish final DNN model	58.1	79.6	92.1

Table 2. Word recognition rates with improved AMs

7. Conclusions

In this paper, we present a series of methods and procedures that allowed us to build a Polish LVCSR system applicable to the automatic broadcast transcription task. We were able to adapt the existing modular ASR platform to a new language within a relatively short time without the need for a dedicated and expensive speech database. The lexicon and the language model were built from public texts available on the Internet and also the acoustic model training used audio data entirely from Polish web pages. The latter was possible due to the iterative schemes that automatically collected, processed, annotated and checked audio data, and trained the AM.

The accuracy we achieved with the best model is fairly good for the target application, which is automatic monitoring of broadcast programs. Most errors that occur (namely in clean speech) are just confusions between similarly sounding word-forms of the same lemma, or omitted very short words (prepositions and conjunctions).

The proposed methods are language independent and we plan to utilize them for other Slavic languages, too.

Acknowledgment

This work was supported by the Technology Agency of the Czech Republic in project no. TA04010199 and by the Student's Grant Scheme (SGS 2015) at TUL.

References

(Demenko et al., 2003) Demenko, G, Wypych, M, Baranowska, E. Implementation of grapheme-to-phoneme rules and extended SAMPA alphabet in Polish text-to-speech synthesis. *Speech and Language Technology* 7 (2003): 79-97.

(Demenko et al., 2008) Demenko, G. et al. JURISDIC: Polish Speech Database for Taking Dictation of Legal Texts. *Proc. of LREC*, 2008.

(Demenko et al., 2012) Demenko, G. et al. Development of Large Vocabulary Continuous Speech Recognition for Polish. Acta Physica Polonica, vol. 121, No. 1-A

(Koržinek and Brocki, 2007) Koržinek, D and Brocki, L. Grammar based automatic speech recognition system for the Polish language. *Recent Advances in Mechatronics*. Springer, 2007, pp. 87-91

(Lööf, J. et al., 2009) Lööf, J, Gollan, C., Ney, H. Cross-language bootstrapping for unsupervised acoustic model training: rapid development of a Polish speech recognition system. *Proc. of Interspeech*, 2009.

(Marasek, 2003) Marasek, K. Large vocabulary continuous speech recognition system for Polish. *Archives of Acoustics,* 28.4, 2003.

(Nouza and Bohac, 2011) Nouza, and Bohac, M. Using TTS for fast prototyping of cross-lingual ASR applications. *Analysis of Verbal and Nonverbal Communication and Enactment*. Springer, 2011, pp. 154-162.

(Nouza et al., 2013) Nouza, J., Cerva, P., Kucharova, M. Cost-Efficient Development of Acoustic Models for Speech Recognition of Related Languages. *Radioengineering*, vol. 22, no. 3, pp. 866-873, 2013

(Nouza et al., 2014) Nouza J, et al. Speech-To-Text Technology to Transcribe and Disclose 100,000+ Hours of Bilingual Documents from Historical Czech and Czechoslovak Radio Archive. *Proc. of Interspeech*, Singapore, 2014, , pp. 964-968

(Pawlaczyk and Bosky, 2009) Pawlaczyk, L. and Bosky, P. Skrybot–a system for automatic speech recognition of polish language. *Man-Machine Interactions*. Springer Berlin Heidelberg, 2009, pp. 381-387.

(Schultz, 2002) Schultz, T.: GlobalPhone: A Multilingual Speech and Text Database developed at Karlsruhe University. *Proc. of ICSLP*, 2002, pp. 345–348.

(Seps et al., 2014) Seps, L. et al. Investigation of Deep Neural Networks for Robust Recognition of Nonlinearly Distorted Speech. *Proc. of Interspeech*, Singapore, 2014.

(Vu et al, 2010a) Vu, N.T. et al.: Rapid bootstrapping of five eastern European languages using the rapid language adaptation toolkit. *Proc. of Interspeech* Makuhari, 2010 pp. 865–868

(Vu et al, 2010b) Vu, N.T., Kraus, F., Schultz, T. Multilingual A-stabil: A new confidence score for multilingual unsupervised training. *Spoken Language Technology Workshop (SLT)*, IEEE, 2010

(Ziółko et al., 2011) Ziółko, M., et al. Automatic speech recognition system dedicated for Polish. *Proc. of Interspeech*, Florence, 2011