Towards Accurate Text Verbalization for ASR Based on Audio Alignment

Diana Geneva and **Georgi Shopov**

IICT - BAS
2, Acad. G. Bonchev Str.
1113 Sofia, Bulgaria
{dageneva,gshopov}@lml.bas.bg

Abstract

Verbalization of non-lexical linguistic units plays an important role in language modeling for automatic speech recognition systems. Most verbalization methods require valuable resources such as ground truth, large training corpus and expert knowledge which are often unavailable. On the other hand a considerable amount of audio data along with its transcribed text are freely available on the Internet and could be utilized for the task of verbalization. This paper presents a methodology for accurate verbalization of audio transcriptions based on phone-level alignment between the transcriptions and their corresponding audio recordings. Comparing this approach to a more general rule-based verbalization method shows a significant improvement in ASR recognition of non-lexical units. In the process of evaluating this approach we also expose the indirect influence of verbalization accuracy on the quality of acoustic models trained on automatically derived speech corpora.

1 Introduction

Automatic speech recognition (ASR) systems transcribe utterances into sequences of linguistic units. Linguistic units can be roughly characterized as either lexical (e.g. “house”, “seven”, “second”) or non-lexical – units that have different verbal and written form (e.g. “11”, “02.07.2017”, “cm”). The form of the linguistic units output from an ASR system depends on the units of the language model (LM). In order for an ASR system to be able to output certain linguistic units their phonetizations have to be known. This poses a problem because most LM training corpora contain both lexical and non-lexical units and while the phonetizations of most of the lexical units can be found in a pronunciation lexicon, this is not the case for the non-lexical units. Most of the methods addressing this issue follow one of two general approaches.

The first approach aims to verbalize the language model training corpus, i.e. to expand all non-lexical units to their verbal forms, and then train a verbal-domain language model on the resulting text that contains only lexical linguistic units (Chelba et al., 2010). Verbalization is often done using finite-state rewrite rules and is a non-trivial task since the choice of correct verbalization is ambiguous as it depends on the context in which the non-lexical unit is used. This approach has several disadvantages. Writing verbalization rules that make use of contextual information is a very time consuming task. It often requires domain specific knowledge and even then in many cases multiple correct verbalizations exist. On the other hand, context-independent rules select a single verbalization variant for each non-lexical unit which is usually very inaccurate because of the aforementioned ambiguities of written language. Alumäe et al. (2017) show how a small amount of verbalized text that serves as ground truth can be used to mitigate the lack of verbalization variability when using context-independent rules. Their method chooses a verbalization for each sentence by sampling from all of its possible verbalization variants with probability that is proportional to the probability of each individual variant. The probability of the variants is assigned by a language model trained on the ground truth text. Nevertheless, the verbalization may still be inaccurate due to the use of sampling.

The second approach is to train a written-domain language model on the original corpus that contains both lexical and non-lexical units and add

Proceedings of the Student Research Workshop associated with RANLP-2019, pages 39–47,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/issn.2603-2821.2019_007

to the pronunciation lexicon the phonetizations of all non-lexical units from the language model vocabulary. Sak et al. (2013) show how this can be implemented without modifying the lexicon. They construct a verbalizer transducer that maps vocabulary items to verbal expansions and compose its inverse with the written-domain language model to produce a verbal-domain language model. This approach, however, is applicable mainly for very large training corpora where the size of the data alleviates the data sparsity issues caused by increasing the size of the language model vocabulary.

The increase of available multimedia content on the Internet and the development of speech and language technology in recent years have made it possible to significantly reduce the manual work needed to prepare speech and language resources. For example, the considerable amount of available audio data along with its transcribed text (such as audiobooks and recorded parliament plenary sessions) has been used for the creation of ASR corpora. The English ASR corpus LibriSpeech (Panayotov et al., 2015) has been derived from thousands of public domain audiobooks. Also, parliament session recordings have been utilized for building ASR corpora for Bulgarian (Geneva et al., in press), Catalan (Miró et al., 2014) and Icelandic (Helgadóttir et al., 2017).

In this paper we describe another way of taking advantage of such resources. We present a methodology for verbalization of audio transcriptions by decoding the corresponding audio with an ASR system and choosing the transcription verbalization that best matches the ASR output phonetically. The idea of using phonetic similarity is quite intuitive because it replicates what a human would do when faced with this task – to determine the correct verbal expansion of a non-lexical unit in a transcription he would have to frequently resort to listening to the audio. Using this methodology we aim to produce more accurate verbalization without the requirements of having large training corpora, ground truth or expert knowledge. Improvements in verbalization accuracy may lead not only to superior language models but could also indirectly improve the quality of acoustic models. Most of the ASR corpora derived from transcribed audio are based on automatic alignment of the audio with its transcription. For such tasks, having a more accurate verbalization method applied to the transcriptions would lead to better alignments and improved quality of the resulting speech corpus.

In the following sections we present the methodology for text verbalization described above and its application to the transcriptions of the plenary sessions of the Bulgarian Parliament. Section 2 describes the data available from the Bulgarian Parliament and the speech corpora and language resources we used to build an acoustic model for ASR. In Section 3 we apply a baseline verbalization method based on rewrite rules to the transcriptions of parliament speeches. In Section 4 we present the method of verbalization based on audio alignment and the process of applying it to the dataset from the Bulgarian Parliament described in Section 2. Finally, in Section 5 we measure the impact on ASR accuracy of the method described in Section 4 in comparison to the method from Section 3. We also provide evidence of the importance of verbalization accuracy to the quality of automatically derived speech corpora.

2 Data Preparation

2.1 Audio and Transcriptions

The website of the Bulgarian Parliament[1] provides video recordings of all plenary sessions since 2010 in mp4 format. The speeches are recorded using stationary directed microphones on the parliament's platform. The format of the audio stream in the video files is 44100 Hz mono compressed with the AAC codec at 75 kb/s. Each recording is separated into parts by the pause breaks made during the session. For each session the corresponding manually transcribed texts are provided in a single file. We downloaded the recordings and their transcriptions from 2010 until July 2018.

The preprocessing of the video files consisted of extracting the audio stream in 16 kHz PCM wav format using the ffmpeg[2] tool. The preparation of the transcriptions had to overcome several specific types of annotations that are present in the text but are not spoken in the audio. We will briefly discuss those issues. Geneva et al. (in press) treat them in more detail.

Every speech in the transcriptions is preceded by the name of the speaker and sometimes the name of his or her party written in free text.

[1] https://www.parliament.bg/
[2] https://ffmpeg.org

The text files also contain annotations that indicate what is happening in the room. All occurrences of both of those annotation types were consistently formatted and contained specific keyphrases which made it easy to construct regular expressions to remove them.

As mentioned above, for each session of the parliament there are several audio files but only one text transcription. We used the semi-automatic approach described by Geneva et al. (in press) to split the transcriptions so that they match the audio session parts. Despite of that, occasional inaccuracies still remain in the alignment of the session audio and text parts.

The resulting dataset consists of 1046 session recordings (2261 parts) for a total of 4832 hours of audio and 30 million words of text.

2.2 Acoustic Model

The Bulgarian ASR corpus BG-PARLAMA (Geneva et al., in press) is a speech corpus built from the speeches of the Bulgarian Parliament members. Its training set consists of 148607 speech segments from 572 unique speakers (422 male and 150 female) with a total duration of 249 hours.

We used the Kaldi ASR Toolkit (Povey et al., 2011) to train a time delay deep neural network (TDNN) (Peddinti et al., 2015) acoustic model with p-norm nonlinearities (Zhang et al., 2014) on the BG-PARLAMA corpus. A speaker-adaptive GMM model was also trained and used for generating state-level alignments for the TDNN training. We used the same parameters for the models as those in the LibriSpeech (Panayotov et al., 2015) Kaldi recipe. The phonetic system that we used is presented in (Mitankin et al., 2009; Hateva et al., 2016) and the pronunciation lexicon is the extended version (Geneva et al., in press) of the lexicon from (Mitankin et al., 2009).

3 Verbalization Based on Rules

In this section we describe the application of the verbalization method based on rewrite rules to the transcriptions from the Bulgarian Parliament. We use it as a baseline for comparison with the verbalization based on audio alignment described in Section 4.

3.1 Rules for Non-Lexical Units

In the transcriptions several frequently occurring types of non-lexical units are observed. They are presented in Table 1 alongside their frequencies and several example occurrences.

The verbalization of some of those units does not require contextual information and is therefore accomplished using simple rewrite dictionaries. The special symbols and some abbreviations fall under this category. Example lines from their rewrite dictionaries are shown below.

$$\S \longrightarrow \text{параграф}$$
$$\text{чл.} \longrightarrow \text{член}$$

There are also non-lexical units that require contextual information to uniquely determine their correct verbalization. In Table 1 only the metric units fall under this category. In general, the verbalization of a metric unit depends on the number preceding it. The singular form is used if the number is "1" and the plural otherwise. For example

$$\text{км.} \longrightarrow \text{километър}/1\ \_$$
$$\text{км.} \longrightarrow \text{километра}/\text{Digit}^* - 1\ \_$$

where $A \rightarrow B/L\_R$ denotes *"replace A with B when the left context is L and the right context is R"*.

The verbalization of the rest of the units from Table 1 (numbers, dates and times) is ambiguous because even though it requires contextual information, the correct verbalization is not uniquely determined by it. We will briefly discuss some of the causes for those ambiguities.

In Bulgarian numbers have cardinal and ordinal forms. Each form has three inflections based on gender (some of which coincide). Often more than one of these forms is a possible verbalization variant. For example, both the ordinal "алинея първа" and the cardinal "алинея едно" are correct verbalizations of "алинея 1". Another source of ambiguity is the fact that some numbers have doublet forms (e.g. "дванадесет" /dvanadeset/ and "дванайсет" /dvanayset/).

In colloquial speech it is common to omit whole parts of phrases. In date expressions the word for "year" is often left out as well as the words for "hours" and "minutes" in expressions for time. For years after 2000 the word for "thousands"("хиляди") is often skipped as in "две и втора" compared to "две хиляди и втора". For

Unit Type	Example Occurrences	Frequency
Arabic numbers	“21”, “42”	865275
Roman numbers	“II”, “XIV”	10200
Fractional numbers	“3.5”, “25,03”	19488
Dates	“07.06.2019”, “27 март 2019”	119209
Abbreviations	“чл.”, “проф.”	208216
Special symbols	“§”, “№”, “+”	105437
Metric units	“км.”, “дка.”, “лв.”	87770
Times	“12 ч. и 23 мин.”, “14,00 ч.”	6229

Table 1: Most frequent non-lexical units found in the transcriptions.

years starting with “19” a shorter form is also accepted such as “деветдесет и четвърта”(ninety-fourth) for “1994”.

In spite of the above-mentioned ambiguities, we verbalized numbers, dates and times using rewrite rules by choosing only one of the possible verbalization variants to expand all of their occurrences.

3.2 Recognition Errors

Before applying the verbalization rules described in the previous subsection the out-of-vocabulary words in the corpus were 4.97% using the extended lexicon from Section 2. After applying them we covered more than 99.65% of the vocabulary found in the transcriptions.

We trained a 3-gram modified Kneser-Ney smoothed language model on the verbalized text using the SRILM Toolkit (Stolcke, 2002). With this language model and the acoustic model described in Section 2 we decoded recordings of the Bulgarian Parliament from 2019 that contain relatively many non-lexical units. In the recognition results we observed systematic mistakes caused by the non-variability in the verbalization. The underlined words in the following snippet demonstrate some of the most common mistakes.

> ... за времето от шест до осемнайсети юни ... гласували сто и едно народни представители ... създават се нова алинея две и алинея три ... десети октомври две хиляди и седемнайсета година ...

The first underlined word is an example of incorrect usage of a cardinal instead of an ordinal number form. The second is a number form that should have agreed on gender with the word that follows it. Even though the cardinal number form in the third and the fourth underlined words is permitted, it differs from the ordinal form that is spoken in the audio recording. The last underlined word should have been omitted because it is not pronounced at all.

We suspect that all those mistakes are caused by the language model. More specifically, because of the lack of variability in the verbalization of its training texts. The method presented in the next section corrects most of those mistakes and thus confirms our assumption.

4 Verbalization Based on Audio Alignment

In this section we present a method for verbalization of audio transcriptions based on phone-level alignment with the audio. Subsection 4.1 presents the creation of a written-domain language model from the transcribed texts and the extension of the pronunciation lexicon with all possible phonetizations of the non-lexical LM vocabulary items. Subsection 4.2 is devoted to the algorithm for phone-level alignment of the ASR output with the audio transcriptions. The algorithm is a modification of the classic algorithm for calculating the Levenshtein distance between strings. It allows to compute the Levenshtein distance between a given string and the concatenation of finite sets of strings. We prove the correctness of this algorithm in Appendix A.

4.1 Building Written-Domain LM and Extending the Lexicon

First, we identify the non-lexical words whose verbalization could not be uniquely determined. We tag them with special tags using rules based on those from the previous section. As seen in Table 1 the occurrences of time expressions are too few so they are treated alongside the unambiguous

non-lexical words as described in Section 3.

We aim to add all possible phonetizations of those specially tagged units to the lexicon. However, this would lead to a major increase in the lexicon size. In order to avoid this we separate some of the tagged expressions into parts and add their phonetizations instead. For example instead of tagging whole date expressions such as "ᴛᴅ02.07.2017ᴛᴅ" we tag the day, month and year separately "ᴛᴅᴅ02ᴛᴅᴅ.ᴛᴅᴍ07ᴛᴅᴍ.ᴛᴅʏ2017ᴛᴅʏ". Non-integer numbers in decimal form are also separated into their integer and fractional parts. The different pronunciations of the decimal separator are also taken into consideration.

In order to reflect the specifics of the language more closely additional tags are introduced. Time periods are tagged separately from ordinary dates because "01–02 юни 2017" could be also pronounced with a "from-to" construction. In some cases a word could be omitted (such as the "year" word in date pronunciations) or a punctuation mark could be pronounced (e.g."dash" and "dot"). Thus new tags were introduced to reflect those specifics. Lastly, acronyms are also tagged separately because they have several pronunciation variants including their expanded form and several different letter-by-letter pronunciations.

The special tags, their frequencies and the verbalization variants that we deemed acceptable are presented in Table 2. We automatically generated the verbalization variants shown in the second column of Table 2 using rewrite rules compiled into finite-state transducers. The verbalizations were then processed using the phonetization rules from the Bulgarian Text-to-Speech System (Andreeva et al., 2005). The phonetization rules require accent information so the accent marks were manually added when needed. This resulted in the expansion of the lexicon with 31935 additional entries.

A 3-gram language model with modified Kneser-Ney smoothing was trained on the resulting tagged text and the updated lexicon. The Kaldi ASR Toolkit was used to decode the downloaded audio from the Bulgarian Parliament (see Section 2) using this language model and the acoustic model described in Section 2.

4.2 Phone-Level Alignment with Variants

The ASR system produces as output a sequence of words along with their recognized phonetizations. Our aim is to align the specially tagged words in the transcribed text with this output in order to obtain their correct phonetizations (and therefore verbalizations). The simplest alignment scheme we could use is based on word-level Levenshtein distance. This technique is expected to align tagged units in the transcription with tagged units in the ASR output. In practice, however, very often an alternative written form is chosen by the ASR system. The reason for this is that the phonetization of a linguistic unit frequently coincides with a combination of the phonetizations of several other units. For example, the phonetization of "ᴛɴ101ᴛɴ" in the transcribed text is expected to be aligned to "ᴛɴ101ᴛɴ" in the ASR output. However, one of the phonetizations of "ᴛɴ101ᴛɴ" (/sto i edno/) coincides with phonetizations of "ᴛɴ100ᴛɴ и ᴛɴ1ᴛɴ" and "сто и едно". Because of this the ASR system could choose any of them and not specifically "ᴛɴ101ᴛɴ". This makes the word-level alignment inappropriate. Thus, we propose the use of phone-level alignment.

Looking at the ASR system output as a sequence of phones we aim to find its best alignment to any of the possible phonetizations of the transcribed text. Each phonetization is formed by the concatenation of possible phonetizations of its constituent words. If $w_1 w_2 ... w_n$ is the transcribed text and $\Phi(w_i)$ is the set of all possible phonetizations of w_i, then the resulting transcription phonetizations are $\Phi(w_1) \circ \Phi(w_2) \circ \cdots \circ \Phi(w_n)$ where $\circ$ denotes concatenation (see Appendix A). This could be used to solve the word-level alignment problem – in the example above, regardless of the word chosen by the ASR system, if the recognized phones are /sto i edno/, then the alignment will select the correct phonetization from Φ("ᴛɴ101ᴛɴ"). The corresponding verbalization could then be uniquely determined from the tagged unit and the chosen phonetization.

The algorithm (see Algorithm 1) is a modification of the Levenshtein distance algorithm (Wagner and Fischer, 1974) and takes into account all phonetization variants for each word in the transcribed text. Given a sequence of phones from the ASR system output $\alpha = a_1 a_2 \ldots a_n$, a sequence of words $\beta = b_1 b_2 \ldots b_m$ that represents a transcription text and a function Φ which yields all possible phonetizations of a given word, the algorithm finds the best alignment between all possible phonetizations of the transcribed text $\Phi(b_1) \circ$

Type	Verbalization Variants	Frequency
TN	doublet forms; ordinal; cardinal; all genders	808836
TRN	doublet forms; ordinal; cardinal; all genders	10194
TFN1	cardinal all genders	19488
TFN2	cardinal all genders; different decimal separator pronunciations	19488
TDD	doublet forms; ordinal masculine; optional leading zero	55765
TDM	doublet forms; ordinal masculine; optional leading zero; month name	2242
TDY	doublet forms; optional "thousands" word	104562
TDYW	optional	102136
TDDPERIOD	TDD variants and optional "from-to" construction	497
TDYPERIOD	TDY variants and optional "from-to" construction	4294
TPUNCT	optional	53298
TAC	expanded forms; different letter-by-letter pronunciations	67931

TN – arabic number, TRN – roman number, TFN1 – integer part of decimal non-integer, TFN2 – fractional part of decimal non-integer, TDD – day, TDM – month, TDY – year, TDYW – year word, TDDPERIOD – time period with dash between days, TDYPERIOD – time perid with dash between years, TPUNCT – some punctuation marks, TAC – abbreviations and acronyms

Table 2: Tagged non-lexical units.

$\Phi(b_2) \circ \cdots \circ \Phi(b_m)$ and α. For each $0 \leq i \leq n$ and $0 \leq j \leq m$ the best alignments between $\Phi(b_1) \circ \Phi(b_2) \circ \cdots \circ \Phi(b_i)$ and $a_1 a_2 \ldots a_j$ are stored in $\mathcal{M}[i][j]$. For each alignment in $\mathcal{M}[i-1]$ we choose the phonetization of b_i which best extends it and write it at the corresponding position in $\mathcal{M}[i]$. This is done in the for loop on line 8. LEVENSHTEINDISTANCE$(\varphi, \alpha, M[i-1])$ fills the dynamic programming table used for the computation of the Levenshtein distance between φ and α. It implements the standard Levenshtein algorithm described in (Wagner and Fischer, 1974). It uses $\mathcal{M}[i-1]$ as a first row, i.e. it extends the best alignments so far. The selection of the best extensions for each prefix of α is done in the for loop on line 10. In the end, $\mathcal{M}[m][n]$ contains the best alignment between $\Phi(b_1) \circ \Phi(b_2) \circ \cdots \circ \Phi(b_m)$ and α. The correctness of the algorithm is further discussed in Appendix A.

The proposed method is then applied to the transcribed texts. Since the agreement between the audio and its transcription is not perfect, we consider the different alignment situations between each tagged unit and the section in the ASR output it's aligned to. If a possible phonetization of the unit exactly matches its aligned section or is a substring of it, then this phonetization is chosen. Otherwise, we choose that phonetization of the unit which is within a given threshold distance (33% phone error rate in our case) to the aligned section, if such exists. If none of those conditions are met, we choose a default phonetization based on the most frequent occurrences of the unit type. In Table 3 the frequency of those choices is shown.

Alignment Type	Frequency
Exact matches	919122
Substring matches	36132
Levenshtein distance $\leq 33\%$	95960
Remaining (default)	197517

Table 3: Frequency of phonetization choices based on the phone-level alignment.

5 Results

During the preliminary tests with the verbalization method from Section 4 we observed that many of the recognition errors described in Section 3 were still present. For example, even though both "алинея едно" and "алинея първа" occur in the language model from Section 4, the ordinal form "алинея първа" was consistently recognized as the cardinal "алинея едно". This lead us to believe that the problem lies within the acoustic model. The texts of the BG-PARLAMA training set contain only occurrences of the cardinal form. We supposed that non-variability in the verbalization used for the preparation of BG-PARLAMA

Algorithm 1 Pseudocode of the phone-level alignment algorithm with variants.

$\mathcal{F} \leftarrow$ phones in the phonetization system
$\mathcal{D} \leftarrow$ language model vocabulary
$\Phi \leftarrow$ function that maps every word in $\mathcal{D}$ to a finite set of its phonetizations
$\alpha \leftarrow$ sequence of phones $a_1 a_2 \dots a_n \in \mathcal{F}^*$ output from an ASR system
$\beta \leftarrow$ sequence of words $b_1 b_2 \dots b_m \in \mathcal{D}^*$ that represents a transcription text
$\mathcal{M} \leftarrow$ an $(m+1) \times (n+1)$ matrix such that $\mathcal{M}[0][j] = j$ for $0 \leq j \leq n$ and $\mathcal{M}[i][j] = \infty$ for $1 \leq i \leq m$ and $0 \leq j \leq n$
for $i \leftarrow 1, m$ **do**
    for all $\varphi \in \Phi(b_i)$ **do**
        $\mathcal{M}' \leftarrow$ LEVENSHTEINDISTANCE$(\varphi, \alpha, \mathcal{M}[i-1])$
        for $j \leftarrow 0, n$ **do**
            $\mathcal{M}[i][j] \leftarrow$ MIN$(\mathcal{M}[i][j], \mathcal{M}'[|\varphi|][j])$
        end for
    end for
end for

lead to mismatches between the audio and its text. In order to test this hypothesis we removed all speeches from the corpus which contain the word "едно" and trained a new TDNN acoustic model with the same parameters. Using this acoustic model and the language model from Section 4 the above-mentioned mistakes were corrected which confirmed the hypothesis. Similar recognition errors caused by the speech corpus were observed between doublet forms of numbers.

Since the number of non-lexical units in the transcriptions is significantly lower than the number of lexical units, we use a similar metric to that in (Sak et al., 2013). Instead of word error rate (WER) we compute non-lexical unit error rate (NER) defined as:

$$\frac{ND + NI + NS}{NN}$$

where NN is the total number of non-lexical number and ND, NI and NS are respectively the number of deletions, insertions and substitutions of non-lexical units. We compared the NER of two ASR systems based on the acoustic model described above that differ only in the language model – the first uses the LM from Section 3, while the second uses the LM from Section 4. The test and dev sets of BG-PARLAMA contain hardly any non-lexical units. This is why the ASR systems were used to decode the specially chosen parliament session from the 5th of June 2019. It contains 758 non-lexical units which we manually transcribed. Examination of the recognition results revealed that many of the mistakes were caused by the system choosing the wrong number doublet form. As we already mentioned, those mistakes are the result of imperfections in the speech corpus. Thus, they should not be included in the verbalization performance comparison. The NER with the first and second LM are shown in Table 4. As it can be seen, the verbalization method presented in Section 4 halved the NER of the verbalization method described in Section 3. Investigation of the recognition errors proved that the alignment-based method is able to correct many of the errors caused by the non-variability of the rule-based method. In order to achieve better estimate of the improvement the aforementioned mismatches present in the speech corpus would have to be reduced.

Verbalization Method	NER
Based on rules	22.8%
Based on alignment	11.5%

Table 4: Non-lexical error rate on the parliament session from the 5th of July 2019.

6 Conclusion

In this paper we described a method for text verbalization based on phone-level alignment between transcriptions and their corresponding audio recordings. We compared it to a general rule-based verbalization method and showed significant reduction in the recognition error rate of non-lexical units. The comparison tests showed that verbalization plays an important role not only in

language modeling but it could indirectly affect the quality of acoustic models as well. We plan to further analyze the mistakes we discovered in the BG-PARLAMA corpus and explore how more accurate verbalization methods could lead to better automatically derived speech corpora.

Acknowledgments

The research presented in this paper is partially funded by the Bulgarian Ministry of Education and Science via National Science Program "Electronic healthcare in Bulgaria" (e-Zdrave) grant DO1-200/2018 and National Science Program "Information and Communication Technologies for Unified Digital Market in Science, Education and Security" grant DO1-205/2018.

References

Tanel Alumäe, Andrus Paats, Ivo Fridolin, and Einar Meister. 2017. Implementation of a Radiology Speech Recognition System for Estonian Using Open Source Software. In *INTERSPEECH*.

Maria Andreeva, Ivaylo Marinov, and Stoyan Mihov. 2005. SpeechLab 2.0: A High-Quality Text-to-Speech System for Bulgarian. In *Proceedings of the RANLP International Conference 2005*. pages 52–58.

Ciprian Chelba, Johan Schalkwyk, Thorsten Brants, Vida Ha, Boulos Harb, Will Neveitt, Carolina Parada, and P. S. Xu. 2010. Query language modeling for voice search. *2010 IEEE Spoken Language Technology Workshop* pages 127–132.

Diana Geneva, Georgi Shopov, and Stoyan Mihov. in press. Building an ASR Corpus Based on Bulgarian Parliament Speeches. In *Proceedings of the SLSP*.

Neli Hateva, Petar Mitankin, and Stoyan Mihov. 2016. BulPhonC: Bulgarian Speech Corpus for the Development of ASR Technology. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. pages 771–774. http://www.lrec-conf.org/proceedings/lrec2016/summaries/478.html.

Inga Rún Helgadóttir, Róbert Kjaran, Anna Björk Nikulásdóttir, and Jón Guonason. 2017. Building an ASR Corpus Using Althingi's Parliamentary Speeches. In *Proc. INTERSPEECH*. pages 2163–2167. https://doi.org/10.21437/Interspeech.2017-903.

Xavier Anguera Miró, Jordi Luque, and Ciro Gracia. 2014. Audio-to-text alignment for speech recognition with very limited resources. In *INTERSPEECH*. pages 1405–1409.

Petar Mitankin, Stoyan Mihov, and Tinko Tinchev. 2009. Large vocabulary continuous speech recognition for Bulgarian. In *Proceedings of the RANLP 2009*. pages 246–250.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pages 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964.

Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. A time delay neural network architecture for efficient modeling of long temporal contexts. In *INTERSPEECH*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Hasim Sak, Françoise Beaufays, Kaisuke Nakajima, and Cyril Allauzen. 2013. Language model verbalization for automatic speech recognition. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* pages 8262–8266.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *INTERSPEECH*.

Robert A. Wagner and Michael J. Fischer. 1974. The String-to-String Correction Problem. *J. ACM* 21(1):168–173. https://doi.org/10.1145/321796.321811.

X. Zhang, J. Trmal, D. Povey, and S. Khudanpur. 2014. Improving deep neural network acoustic models using generalized maxout networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pages 215–219. https://doi.org/10.1109/ICASSP.2014.6853589.

A Correctness of the Alignment with Variants

We will make use of some standard terms from formal language theory. We call an alphabet any finite set of symbols. A string over some alphabet is a finite sequence of symbols from that alphabet. With $|\alpha|$ we denote the length of a string α, i.e. the length of the corresponding sequence. We will use ε to denote the unique string of length 0 and Σ^* to denote the set of all strings over the alphabet Σ. The operation "concatenation of strings" and its lifted version to sets of strings will be denoted with $\circ$. That is for the strings $\alpha = a_1 a_2 \dots a_n$ and

$\beta = b_1 b_2 \dots b_m, \alpha \circ \beta = a_1 a_2 \dots a_n b_1 b_2 \dots b_m$ and for the sets of strings A and B, $A \circ B = \{\alpha \circ \beta \mid \alpha \in A \wedge \beta \in B\}$.

The Levenshtein distance between two strings s_1 and s_2 is defined as the minimum number of operations (insertions, deletions and substitutions) that transform s_1 into s_2. This can be formalized as follows.

Definition 1. Let Σ be an alphabet. We define the set

$$op(\Sigma) \coloneqq \{(a, b) \mid a, b \in \Sigma \cup \{\varepsilon\} \wedge (a, b) \neq (\varepsilon, \varepsilon)\},$$

and the function $w\colon op(\Sigma) \to \{0, 1\}$ as $w((a, b)) = 0$ iff $a = b$, for any $(a, b) \in op(\Sigma)$.

Definition 2. Let Σ be an alphabet and $\alpha, \beta \in \Sigma^*$. An *alignment* of α and β is a string $\gamma \in op(\Sigma)^*, \gamma = (a_1, b_1)(a_2, b_2) \dots (a_n, b_n)$ such that $\alpha = a_1 \circ a_2 \circ \dots \circ a_n$ and $\beta = b_1 \circ b_2 \circ \dots \circ b_n$. The *weight* of γ is $\widehat{w}(\gamma) = \sum_{i=1}^{n} w((a_i, b_i))$. We use $ali(\alpha, \beta)$ to denote the set of all alignments of α and β.

Definition 3. The Levenshtein distance between the strings $\alpha \in \Sigma^*$ and $\beta \in \Sigma^*$ is defined as

$$lev(\alpha, \beta) \coloneqq \min\{\widehat{w}(\gamma) \mid \gamma \in ali(\alpha, \beta)\}.$$

Definition 4. The Levenshtein distance between a string $\alpha \in \Sigma^*$ and a set of strings $B \subseteq \Sigma^*$ is defined as

$$\widehat{lev}(\alpha, B) \coloneqq \min \bigcup_{\beta \in B} \{\widehat{w}(\gamma) \mid \gamma \in ali(\alpha, \beta)\}.$$

In our case we have an alphabet $\mathcal{F}$ – the phones in the phonetization system, an alphabet $\mathcal{D}$ – the words in the LM vocabulary, and a function $\Phi\colon \mathcal{D} \to \mathcal{P}(\mathcal{F}^*)$ which maps every word in $\mathcal{D}$ to a finite set (the phonetizations of the word). Given the phone output of the ASR system $\alpha \in \mathcal{F}^*, \alpha = a_1 a_2 \dots a_n$, and $\beta \in \mathcal{D}^*$, $\beta = b_1 b_2 \dots b_m$ – the words in the transcription text, we look for the Levenshtein distance between α and the set $\Phi(b_1) \circ \Phi(b_2) \circ \dots \circ \Phi(b_m)$. We will use α_i and β_i to denote the prefixes of respectively α and β of length i. We will also write $\Phi(\beta_i)$ instead of $\Phi(b_1) \circ \dots \circ \Phi(b_i)$. As already mentioned, the LEVENSHTEINDISTANCE function from Algorithm 1 implements the standard Levenshtein algorithm using a predefined first row for the dynamic programming table. Its correctness follows directly from the correctness of the Levenshtein algorithm and is expressed in Proposition 1.

Proposition 1. *Let* $\varphi \in \Phi(b_{i+1})$. *If* $\mathcal{M}' = \text{LEVENSHTEINDISTANCE}(\varphi, \alpha, X)$ *where* $X[j] = \widehat{lev}(\alpha_j, \Phi(\beta_i))$ *for* $0 \leq j \leq n$, *then* $\mathcal{M}'[|\varphi|][j] = \widehat{lev}(\alpha_j, \Phi(\beta_i) \circ \{\varphi\})$.

Proof. Straightforward induction on $|\varphi|$. □

In order to demonstrate the correctness of Algorithm 1, i.e. to show that $\mathcal{M}[m][n]$ is the Levenshtein distance between α and $\Phi(\beta_m)$, it is enough to prove the following proposition.

Proposition 2. *For every* $0 \leq i \leq m$ *at the end of the* $i-th$ *iteration of the for loop beginning on line* 7

$$(\forall 0 \leq j \leq n)(\mathcal{M}[i][j] = \widehat{lev}(\alpha_j, \Phi(\beta_i))),$$

where for $i = 0$ *we assume that* $\Phi(\beta_i) = \{\varepsilon\}$.

Proof. We will prove it by induction on i.

For $i = 0$ the proposition becomes

$$\begin{aligned}(\forall 0 \leq j \leq n)(\mathcal{M}[0][j] &= \widehat{lev}(a_1 a_2 \dots a_j, \{\varepsilon\}) \\ &= lev(a_1 a_2 \dots a_j, \varepsilon)).\end{aligned}$$

Since $\mathcal{M}[0][j] = j$ for $0 \leq j \leq n$ as defined on line 6 and $lev(a_1 a_2 \dots a_j, \varepsilon) = j$ by definition, the base case holds. Let the proposition hold for some $0 \leq i \leq m - 1$. Let $\varphi \in \Phi(b_{i+1})$. Proposition 1 implies that $\mathcal{M}'[|\varphi|][j] = \widehat{lev}(\alpha_j, \Phi(\beta_i) \circ \{\varphi\})$. The for loop on line 10 takes the minimum for each j. Therefore

$$\begin{aligned}\mathcal{M}[i+1][j] &= \min_{\varphi \in \Phi(b_{i+1})} \mathcal{M}'[|\varphi|][j] \\ &= \min_{\varphi \in \Phi(\beta_{i+1})} \widehat{lev}(\alpha_j, \Phi(\beta_i) \circ \{\varphi\}) \\ &= \min_{\varphi \in \Phi(\beta_{i+1})} \bigcup_{\lambda \in \Phi(\beta_i) \circ \{\varphi\}} \{\widehat{w}(\gamma) \mid \gamma \in ali(\alpha_j, \lambda)\} \\ &= \min \bigcup_{\lambda \in \Phi(\beta_{i+1})} \{\widehat{w}(\gamma) \mid \gamma \in ali(\alpha_j, \lambda)\} \\ &\overset{\text{def}}{=} \widehat{lev}(\alpha_j, \Phi(\beta_{i+1})).\end{aligned}$$

□