ON THE USE OF GRAPHEME MODELS FOR SEARCHING IN LARGE SPOKEN ARCHIVES

Jan Švec[1], *Josef V. Psutka*[2], *Jan Trmal*[3], *Luboš Šmídl*[2], *Pavel Ircing*[2], *Jan Sedmidubsky*[4]

[1]NTIS, [2]Department of Cybernetics, University of West Bohemia, Pilsen, Czech Republic
[3]Center for Language and Speech Processing, Johns Hopkins University, USA
[4]Faculty of Informatics, Masaryk University, Brno, Czech Republic

[honzas,psutka_j,smidl]@kky.zcu.cz, jtrmal@gmail.com, xsedmid@fi.muni.cz

ABSTRACT

This paper explores the possibility to use grapheme-based word and sub-word models in the task of spoken term detection (STD). The usage of grapheme models eliminates the need for expert-prepared pronunciation lexicons (which are often far from complete) and/or trainable grapheme-to-phoneme (G2P) algorithms that are frequently rather inaccurate, especially for rare words (words coming from a different language). Moreover, the G2P conversion of the search terms that need to be performed on-line can substantially increase the response time of the STD system. Our results show that using various grapheme-based models, we can achieve STD performance (measured in terms of ATWV) comparable with phoneme-based models but without the additional burden of G2P conversion.

Index Terms— Spoken term detection, speech indexing, grapheme-based speech recognition, keyword search

1. INTRODUCTION

A spoken term detection (STD) in large spoken document collections is a specific task, especially when the search phase needs to be interactive through some low-latency graphical user interface [1]. In this case, there are only soft limitations on the computational power needed to pre-process the collection. This allows using more complicated structure of the pre-processing pipeline, e.g.: more complicated automatic speech recognition (ASR) models [2], multiple ASR models [3] and speech pre-indexing [4]. On the other hand, there is a strict demand on the fast response from the user interface during the search phase. The user also expects that the search process is able to retrieve any spoken term or phrase, so that not only in-vocabulary (IV) terms but also out-of-vocabulary (OOV) terms have to be searched for. While the IV terms could be easily indexed in the inverted index, the OOV terms are not a priori known and to speed-up the search process other constituent units have to be indexed. Two state-of-the-art methods employ two different types of such units:

- *Sub-word units* such as syllables or phoneme n-grams, which in combination compose the OOV term [4, 5]; and
- *Proxy words* that suppose each OOV term is recognized as a sequence of IV terms [6].

Sub-word units require to maintain a separate inverted index, but also the proxy words need to store the structural properties of the original word lattice to catch the exact sequences of proxy words. The sub-word units could be easily indexed using many freely available database engines. The method based on proxy-words is more demanding because it depends heavily on the use of weighted finite state automatons (WFSA). Although the index of the whole collection could be represented as a special kind of WFSA [7], it requires to recompute and optimize the index WFSA when adding additional records into a collection.

Grapheme-based models are becoming widely used in many applications of speech recognition [8, 9, 10, 11, 12], especially in under-resourced tasks. Their use simplifies the development of the speech recognizer because the rather complicated step of phonetic transcription could be skipped. This is still true even if the grapheme-to-phoneme (G2P) methods (such as Phonetisaurus [13] or Sequitur [14]) are used. The limitations of G2P methods come mainly from the fact that they are also machine-learning methods with a limited accuracy, especially for OOV words. Such words are often words with non-systematic pronunciation (e.g. coming from different languages), yielding the machine-learning methods essentially powerless.

In this paper, we focus on the use of grapheme-based speech recognition models in the domain of large spoken archives. In the experiments, we used the USC-SFI MALACH archive of interviews with Holocaust survivors. We used testimonies in two languages – English [15] and Czech [16]. For both languages, the training data for acoustic and language models are available together with the additional pronunciation lexicons. Especially the English collection contains records of non-native speakers mentioning names and locations with uncertain or irregular English pronunciation (typical examples: German word *führer*, name of Slovakian town *Kežmarok* or Jewish name *Lejerowisz*).

For all such names and terms, the expert-defined pronunciations are specified as part of the English and Czech corpora. The training of G2P in such cases is possible [17] but difficult, since the collection contains a mix of pure English words together with Central European topography and proper names and with German colloquial words and slang related to Holocaust. Although the deficiencies of G2P could be partially alleviated with the ASR language model, this is not true for the OOV words, where the proper transcription of the graphemes to phonemes has to be known to find the word in the search index.

In this context, we wanted to study the effect of using the grapheme-based models in the STD task over the large audio collections, such as USC-SFI MALACH. The goal of the experiments was to clarify the effect of the expert-created pronunciation lexicons used to build phoneme-based models in comparison with the lexicon-free grapheme-based models. We focused on the evaluation of speech recognition error rates as well as on the evaluation of search performance.

2. GRAPHEME-BASED SPEECH RECOGNITION

The grapheme-based speech recognition models differ from the phoneme-based model in the set of context-dependent units. We used direct mapping of graphemes to the context-dependent recognition units as in [10]. The numbers of different graphemes and phonemes for English and Czech are summarized in Tab. 1. For grapheme-based models, we use exactly one grapheme sequence for each word in the recognition lexicon.

To train the acoustic models, we used a typical Kaldi [18] training recipe for a DNN-based training. The same recipe was used for grapheme- and for phoneme-based acoustic models. This recipe uses layer-wise RBM pre-training, stochastic gradient descent training and sequence-discriminative training optimizing sMBR criterion. We used the topology with 5 hidden layers (each with 2,048 neurons) and a softmax output layer. We used features based on standard 12-dimensional Cepstral Mean Normalized (CMN) PLP coefficients with first and second derivatives. We trained two sets of acoustic models – (1) the baseline using the phonemes as context-dependent units with phonetic transcription generated from the pronunciation lexicons and (2) the grapheme-based models using just the graphemes of the lexicon words.

The language model (LM) for both phoneme- and grapheme-based ASR was the standard word trigram LM. The same LM was used in phoneme- and grapheme- experiments. To experiment with the search methods based on sub-word units, we also used 5-gram phoneme and grapheme language models. To recognize the collection, we used our in-house real-time decoder both for the word- and phoneme recognition with trigram word LM and 5-gram phoneme LM. The evaluation of error rates on both the word and sub-word (phonemes/graphemes) levels were evaluated, the results are shown in Tab. 2.

3. SPOKEN TERM DETECTION

Based on the previous work [4, 6, 19], we focus on the use of sub-word units for spoken term detection. Nevertheless, in the experimental part of this work, we compare the method based on sub-words units with the approach of using proxy words in the STD task.

The STD for large spoken archives can be divided into three steps, where the first step is performed off-line. The subsequent two steps are executed after the searched terms are known and the speed of these steps affects the overall responsiveness of the interactive STD system. These steps are:

1. Speech indexing including automatic speech recognition, lattice generation and lattice indexing. The goal of speech indexing is to speed-up the search process by pre-processing the collection of records and storing the information needed to retrieve the searched term in the index.

2. Putative hits detection is the first step of the search phase, where the list of possible candidates (putative hits) of the searched term or phrase is constructed based on the pre-computed index. This step influences the recall of the resulting system because words not occurring in the list of putative hits are definitively missing in the list of results.

3. Term relevance estimation as the second step of the search phase assigns the estimate of the posterior probability that the given putative hit of the given term occurs in the given time interval of the audio record in the collection. This step determines the precision of the system - it scores the putative hits to distinguish between true positive hits and false negative hits. In our experiments, we use the recognized grapheme (or phoneme) confusion network to represent the pre-processed audio records in the collection.

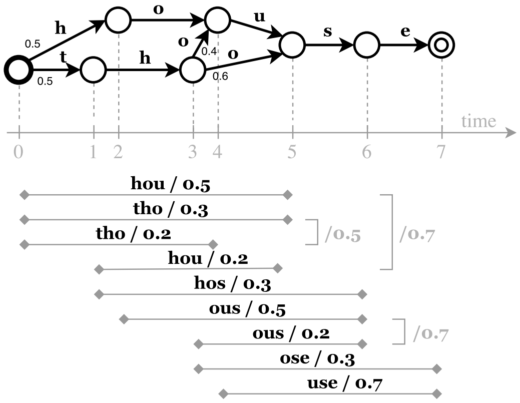

Fig. 1. Indexation of grapheme lattice. The grapheme trigrams with the assigned time intervals are stored in the inverted sub-word index. The (merged) posterior probabilities are used to filter the low-probability trigrams.

3.1. Speech indexing

During speech indexing, we maintain two separate inverted indices, one for searching IV terms and one for OOV terms. The inverted index of IV terms is generated from the word-level ASR lattices, where each record consists of the quintuplet (*word, probability, audio_id, start, end*), where *probability* is the posterior probability of a *word* occurring in *audio_id* at time interval [*start, end*].

To search the OOV terms, we construct the inverted index for sub-word units. For the purpose of this paper, we experimented with n-grams of phonemes and n-grams of graphemes. Based on the previous experiments, we decided to use $n = 3$. This is because the number of different trigrams of phonemes/graphemes is relatively small (for English graphemes there is at most 26^3 different trigrams, but in practice the number is significantly lower) and at the same time the trigrams are specific enough – the query combined from multiple trigrams often leads to a specific occurrence in the audio collection with a acceptable number of false positives. A large number of false positives could be eliminated in the term relevance estimation step. On the other hand, this step negatively affects the search speed, because the score of each false positive has to be estimated and then discarded from the list of results (due to its low posterior probability).

To generate the sub-word index, we first convert the recognized phoneme/grapheme lattice into a factor automaton [20]. The factor automaton encodes the posterior probabilities of all subpaths (n-grams of all orders) in the original lattice. Then, only the subpaths of length $n = 3$ are selected. The overlapping subpaths bearing the same trigram are merged together so that the time span is a union of the two overlapping intervals and the posterior probability is the sum of the two partial probabilities. To keep the index size within a feasible size, we then apply the threshold (in the experiments, we use the value 10^{-4}) on posterior probabilities, so that the trigrams with lower probabilities are discarded and not included in the sub-word index. The whole process is illustrated in Fig. 1.

3.2. Putative hits detection

In the search phase, the workflow depends on the searched term – the IV terms are searched directly in the IV inverted index. An OOV term has to be decomposed into a sequence of sub-word units appropriate the OOV inverted index (i.e. trigrams of graphemes/phonemes). At this point, one can see the obvious advantage of the grapheme-based models and STD, because the OOV terms do not need to

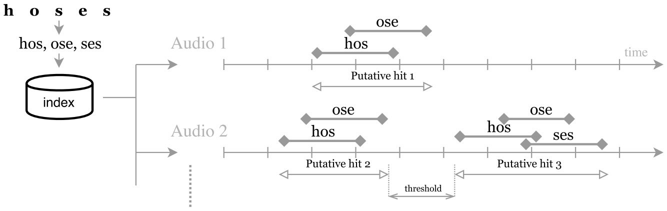

Fig. 2. Searching using sub-word units (grapheme trigrams). The searched term *"hoses"* is decomposed into a sequence of grapheme trigrams and looked up in the inverted index. The exact match is not necessary to indicate the putative hit.

be transcribed into a sequence of phonemes, which is a non-trivial process.

The trigrams of graphemes/phonemes are then looked up in the sub-word index. The results are grouped according to the audio record identifier and sorted according to the time of the occurrence. Then, the putative hits are determined as the clusters of trigrams on the time axis. Two consecutive clusters have to be separated at least by a given threshold (in our experiments, we used 0.3 seconds). Illustration of the putative hits detection is shown in Fig. 2.

The putative hit is not required to be an exact match of all searched trigrams. If the trigrams at the beginning or ending of the word are not matched, the putative hit interval is extended in this direction by a fixed time interval – the reason is that the term relevance estimation could still exploit the information stored in the corresponding part of grapheme/phoneme confusion network.

3.3. Term relevance estimation

To assign the posterior probability to a given putative hit, we used the approach described in [19, 21] based on Siamese neural networks [22]. The approach uses two jointly trained neural networks to distinguish between the same and different training examples. The difference from common machine-learning posterior score estimation methods is that the neural network is not forced to output 0 or 1 for different or same examples. Instead, the inputs are mapped into an embedding vector of fixed dimensionality and the similarity is computed as a cosine distance of two vectors in this output embedding space [23].

Applied to the term relevance estimation, the goal is to assign a relevance score to some segment of the input audio (represented by corresponding part of the recognized grapheme/phoneme confusion network $\hat{x}_w$) given a searched term w (represented by grapheme sequence x_w). Because both x_w and $\hat{x}_w$ are sequences of variable lengths, we use two recurrent neural networks (RNNs) to process x_w and $\hat{x}_w$, respectively. The first RNN $g(\hat{x}_w)$ computes the output embedding $\hat{y}_w$ from the recognized grapheme/phoneme confusion networks and the second RNN $f(x_w)$ computes the output embedding y_w from the graphemes of the searched term w. Finally, the relevance score is estimated as a cosine similarity of the pronunciation embedding $f(x)$ obtained from the graphemes of the query x and the pronunciation embedding $g(\hat{x})$ computed from the recognized grapheme/phoneme confusion network, formally as $d(y_w, \hat{y}_w) = 1 - \cos(f(x), g(\hat{x}))$. For more details, see [19].

Table 1. Statistics of development and test datasets.

	English		Czech	
	Dev	Test	Dev	Test
LVCSR vocabulary	22,723		252,082	
# of graphemes	26		39	
# of phonemes	38		41	
#speakers	10	10	10	10
OOV rate	1.0%	0.7%	3.2%	2.6%
#IV terms	710	735	1762	1764
#OOV terms	154	78	1251	1090
dataset length [hours]	11.1	11.3	20.4	19.4

Table 2. Recognition error rates in % (Grphm. - grapheme based model, Phnm. - phoneme based model).

		Dev data		Test data	
		Grphm.	Phnm.	Grphm.	Phnm.
English	words	26.16	25.39	21.21	20.80
	sub-words	23.51	22.15	23.18	21.33
Czech	words	27.66	23.98	23.12	19.11
	sub-words	20.36	19.21	16.51	16.13

The training data for the Siamese neural networks consist of a set of pairs $(x_w, \hat{x}_w)$ extracted from the large audio collection in the unsupervised fashion. First, the audio collection is recognized using the word- and sub-word- level recognizers. Then, the recognized words with confidences higher than some predefined threshold (in our experiments 0.9) are used as words x_w and the corresponding parts of the grapheme/phoneme confusion networks are used as $\hat{x}_w$. This way, no labeled data nor human labor are required to prepare the training data. During training the Siamese neural network, first the pair of two different words $(w, \bar{w})$ must be sampled from the training data. To model the variations in pronunciation of words, the corresponding grapheme/phoneme confusion networks $\hat{x}_w$ and $\hat{x}_{\bar{w}}$ are sampled from the training set of pairs. Then, the Siamese neural network is trained to optimize the criterion for different pairs $(w, \bar{w})$:

$$l(w,\bar{w}) = \frac{1}{2} \cdot \big(\max\{0, m + d(f(x_w), g(\hat{x}_w)) - d(f(x_w), g(\hat{x}_{\bar{w}}))\} \\ + \max\{0, m + d(f(x_{\bar{w}}), g(\hat{x}_{\bar{w}})) - d(f(x_{\bar{w}}), g(\hat{x}_w))\}\big)$$

To normalize the output scores, we used a simple method of rank normalization with mapping the rank back to posterior probabilities, as described in [24].

4. EXPERIMENTAL RESULTS

We compared the grapheme-based models with their phoneme-based counterparts on the USC-SFI MALACH collection of interviews with Holocaust survivors [15, 16]. We used the English and Czech subset of the collection. The development and test data partitions are summarized in Tab. 1. The RNNs used for term relevance score estimation were trained from 100 hours for each language. The data used to generate the training examples for RNNs were different from the development and test datasets.

ASR evaluation. The first results are from the evaluation of recognition error rates. We trained the grapheme- and phoneme-based models and we used such models to recognize the development and test dataset at the word level and also on the sub-word level (graphemes or phonemes). The results are summarized in Tab. 2. The direct comparison of models on both languages shows a slightly worse performance of the grapheme-based models on both word- and sub-word- levels. It is probably caused not by the lack of phonetic information but rather by the lack of additional knowledge about the pronunciation of irregular words (see Sec. 1 for examples).

STD evaluation. The next set of experiments shows the results of the spoken term detection. We used automatically generated sets of terms in the evaluation. The terms were automatically selected from the graphemic representation of words and the same set was used in phoneme-based and grapheme-based experiments. Each term included in the term set satisfies the following conditions: (1) it has more than three graphemes, (2) the sequence of graphemes is not a subsequence or near-subsequence of another term.

To evaluate the performance, we used the ATWV metric [25]. The optimal decision threshold was determined to maximize the ATWV on the development set and the optimal thresholds were applied to the test set. The results are reported in Tab. 3.

The first conclusion from Tab. 3 shows that the performance of in-vocabulary (IV) search is better for phoneme-based models in English for both the development and test data, but in Czech, the results are comparable. This is probably caused by the fact that the graphemic and phonetic representations of regular Czech words are close to each other.

The next rows of Tab. 3 compare two different methods for out-of-vocabulary (OOV) search: the use of sub-word units and the use of proxy words. The results on OOV terms clearly prefer the method based on sub-word units – the ATWV is significantly higher in comparison to the method based on proxy words for both the grapheme- and phoneme-based models. Also the combined search (the IV terms are searched in word index and OOV terms are searched using sub-word units or proxy words, in Tab. 3 denoted as IV+OOV) shows a preference for the use of sub-word units for speech indexing, especially for the Czech language regardless of the use of graphemes or phonemes for speech recognition.

As for IV terms, the ATWV scores for grapheme- and phoneme-based models are very similar for Czech. The ATWV values for the OOV terms on English are a bit noisy, which could be caused by a smaller number of OOV terms in comparison with the Czech language. The scores for combined search (IV+OOV) are higher for the phoneme-based model on English, it is caused mainly by the lower ATWV scores for IV terms.

Grapheme-mapped word index. The last set of experiments focuses on the use of graphemes only during the putative hit detection. As has been said, the putative hit detection is performed in real-time and its speed affects the overall perceived “snappiness” of an interactive STD system. Additional step of producing the pronunciation using G2P systems, such as Sequitur, slows down the system response. The increase in processing time is caused by two effects: (1) the overall G2P generation time and (2) multiple pronunciation variants for a searched term. Therefore, we experimented with the so-called *grapheme-mapped word index* which is generated from the ASR lattices at the word level. The fact whether the ASR system producing the lattices was phonetic or graphemic does not play any role. The lattices are first converted to the grapheme lattices by replacing the lattice word transitions with a sequence of grapheme transitions. At this point, we would like to point out that the time alignment of the lattices nodes does not have to be precise, because the grapheme time alignment is used only to detect overlapping sub-word units (see Sec. 3.1). Therefore for each word transition, we generated equidistantly spaced states for inserted grapheme transitions. The transition probabilities were adopted from the word lattice.

Then, the index based on sub-word units is generated from these artificially generated grapheme lattices. During the putative hit detection, the searched term is treated as a sequence of graphemes and searched in the sub-word index. It completely eliminates the G2P from the search phase, because the term relevance estimation assigns the posterior probability based on the grapheme representation of the searched term and the recognized grapheme/phoneme confusion network of the putative hit.

The results of experiments with grapheme-mapped word index are shown in Tab. 4. The main observation is that the performance of grapheme-mapped word index STD applied to phoneme-based recognition models is between the pure phoneme-based STD and the grapheme-based STD. In other words, we claim that the sub-word index for putative hit detection could be constructed from the word-level ASR lattices. This way, it is similar to the method based on proxy words, but during the putative hit detection, it is not necessary to obtain the exact match – only the partial match of sub-word units is sufficient to indicate the putative hit.

Table 3. Spoken term detection performance (ATWV metrics) for in-vocabulary (IV) terms, out-of-vocabulary (OOV) terms and combination (IV+OOV).

		Dev data		Test data	
	Searched terms	Grphm.	Phnm.	Grphm.	Phnm.
English	IV	0.7759	0.7970	0.6991	0.7447
	OOV sub-word	0.4176	0.3808	0.2677	0.3799
	IV+OOVsub-word	0.6912	0.7070	0.6394	0.7042
	OOV proxy	0.2481	0.2706	0.2105	0.3080
	IV+OOV proxy	0.6804	0.7005	0.6506	0.6992
Czech	IV	0.8227	0.8224	0.8202	0.8277
	OOV sub-word	0.6644	0.6591	0.6777	0.6818
	IV+OOVsub-word	0.7541	0.7546	0.7621	0.7723
	OOV proxy	0.3163	0.4942	0.3353	0.5031
	IV+OOV proxy	0.6125	0.6905	0.6350	0.7090

Table 4. Spoken term detection performance with word-mapped grapheme index, the IV column is the same as the IV rows in Tab. 3 (ATWV metrics).

		IV	OOV	IV+OOV
English	Graphemes	0.6991	0.2677	0.6394
	+ grph.-mapped index		0.4417	0.6542
	Phonemes	0.7447	0.3799	0.7042
	+ grph.-mapped index		0.3623	0.6895
Czech	Graphemes	0.8202	0.6777	0.7621
	+ grph.-mapped index		0.6260	0.7436
	Phonemes	0.8277	0.6818	0.7723
	+ grph.-mapped index		0.6707	0.7699

5. CONCLUSION

The paper presents a comparison of grapheme- and phoneme-based speech recognition models evaluated on large spoken collections in two languages, English and Czech. The performance of both types of models was similar for Czech. For English, the grapheme-based models performed slightly worse. We also introduced the method of word-mapped index which allows indexing the sub-word units based only on the recognized word lattices. This allows to completely eliminate the G2P algorithm from the search phase of the STD at the price of a small decrease in STD performance.

6. ACKNOWLEDGEMENT

This research was supported by the Grant Agency of the Czech Republic, project No. GAČR GBP103/12/G084. Jan Trmal was supported by the NSF grant No CRI-1513128.

7. REFERENCES

[1] Petr Stanislav, Jan Švec, and Pavel Ircing, "An engine for online video search in large archives of the holocaust testimonies," in *Proc. Interspeech 2016*. 2016, International Speech Communication Association.

[2] X. Chen, A. Ragni, J. Vasilakes, X. Liu, K. Knill, and M. J. F. Gales, "Recurrent neural network language models for keyword search," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 5775–5779.

[3] J. van Hout, L. Ferrer, D. Vergyri, N. Scheffer, Y. Lei, V. Mitra, and S. Wegmann, "Calibration and multiple system fusion for spoken term detection using linear logistic regression," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 7138–7142.

[4] Josef Psutka, Jan Švec, Josef V. Psutka, Jan Vaněk, Aleš Pražák, Luboš Šmídl, and Pavel Ircing, "System for Fast Lexical and Phonetic Spoken Term Detection in a Czech Cultural Heritage Archive," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2011, no. 1, pp. 10, 2011.

[5] C. van Heerden, D. Karakos, K. Narasimhan, M. Davel, and R. Schwartz, "Constructing sub-word units for spoken term detection," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 5780–5784.

[6] Guoguo Chen, Oguz Yilmaz, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013 - Proceedings*, 2013, pp. 416–421.

[7] Dogan Can and Murat Saraclar, "Lattice Indexing for Spoken Term Detection," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 8, pp. 2338–2347, 2011.

[8] Mirjam Killer, Sebastian Stuker, and Tanja Schultz, "Grapheme based speech recognition," in *Proc. Interspeech 2013*, 2013, pp. 3141–3144.

[9] Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication*, vol. 56, no. Supplement C, pp. 85 – 100, 2014.

[10] Jan Trmal, Matthew Wiesner, Vijayaditya Peddinti, Xiaohui Zhang, Pegah Ghahremani, Yiming Wang, Vimal Manohar, Hainan Xu, Daniel Povey, and Sanjeev Khudanpur, "The Kaldi OpenKWS system: Improving low resource keyword search," in *Proc. Interspeech 2017*, 2017, pp. 3597–3601.

[11] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4805–4809.

[12] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec 2017, pp. 193–199.

[13] Josef Robert Novak, Nobuaki Minematsu, and Keikichi Hirose, "Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework," *Natural Language Engineering*, vol. 22, no. 6, pp. 907–938, 2016.

[14] Maximilian Bisani and Hermann Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434 – 451, 2008.

[15] Bhuvana Ramabhadran, Samuel Gustman, William Byrne, Jan Hajič, Douglas Oard, J. Scott Olsson, Michael Picheny, and Josef Psutka, "USC-SFI MALACH Interviews and Transcripts English LDC2012S05," 2012.

[16] Josef Psutka, Vlasta Radová, Pavel Ircing, Jindřich Matoušek, and Luděk Müller, "USC-SFI MALACH Interviews and Transcripts Czech LDC2014S04," 2014.

[17] Aliya Deri and Kevin Knight, "Grapheme-to-phoneme models for (almost) any language," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 399–408, Association for Computational Linguistics.

[18] 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, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[19] Jan Švec, Josef V. Psutka, Luboš Šmídl, and Jan Trmal, "A relevance score estimation for spoken term detection based on rnn-generated pronunciation embeddings," in *Proc. Interspeech 2017*, 2017, pp. 2934–2938.

[20] Mehryar Mohri, Pedro Moreno, and Eugene Weinstein, "Factor automata of automata and applications," *Implementation and Application of Automata*, vol. 4783, pp. 168–179, 2007.

[21] Jan Švec, Luboš Šmídl, and Josef V. Psutka, *An Analysis of the RNN-Based Spoken Term Detection Training*, pp. 119–129, Springer International Publishing, Cham, 2017.

[22] Wanjia He, Weiran Wang, and Karen Livescu, "Multi-view Recurrent Neural Acoustic Word Embeddings," *Appearing in ICLR 2017*, pp. 1–12, nov 2017.

[23] Herman Kamper, Weiran Wang, and Karen Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016, vol. 2016-May, pp. 4950–4954.

[24] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen, J. Makhoul, F. Grezl, M. Hannemann, M. Karafiat, I. Szoke, K. Vesely, L. Lamel, and V. B. Le, "Score normalization and system combination for improved keyword spotting," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, vol. 26, pp. 210–215.

[25] Jonathan G Fiscus, Jerome Ajot, John S Garofolo, and George Doddington, "Results of the 2006 spoken term detection evaluation," in *Proceedings of the ACM SIGIR Conference*, 2007, vol. 7, pp. 51–57.