Leveraging Text Data for Word Segmentation for Underresourced Languages

Thomas Glarner\textsuperscript{1}, Benedikt Boenninghoff\textsuperscript{2}, Oliver Walter\textsuperscript{1}, Reinhold Haeb-Umbach\textsuperscript{1}

\textsuperscript{1}Department of Communications Engineering – Paderborn University, 32100 Paderborn, Germany
\textsuperscript{2}Cognitive Signal Processing Group – Ruhr-Universität Bochum, 44801 Bochum, Germany

glarner@nt.upb.de, benedikt.boenninghoff@rub.de, walter@nt.upb.de, haeb@nt.upb.de

Abstract

In this contribution we show how to exploit text data to support word discovery from audio input in an underresourced target language. Given audio, of which a certain amount is transcribed at the word level, and additional unrelated text data, the approach is able to learn a probabilistic mapping from acoustic units to characters and utilize it to segment the audio data into words without the need of a pronunciation dictionary. This is achieved by three components: an unsupervised acoustic unit discovery system, a supervisedly trained acoustic unit-to-grapheme converter, and a word discovery system, which is initialized with a language model trained on the text data. Experiments for multiple setups show that the initialization of the language model with text data improves the word segmentation performance by a large margin.

Index Terms: underresourced speech recognition, nonparametric Bayesian estimation, unsupervised word segmentation, unsupervised learning

1. Introduction

Automatic speech recognition (ASR) has made great progress in recent years, and ASR systems have been reported to be on par with human transcribers on certain tasks, such as conversational speech recognition [1]. However, these systems rely on supervised learning techniques requiring a significant amount of transcribed speech, language model training data, and a pronunciation dictionary. While these resources are available for all major languages, these major languages form only a few percent of all languages worldwide. For many of the less common languages these resources are not available and simply too expensive to create. In particular, the creation of the pronunciation lexicon is copious since it requires a great amount of linguistic expertise. On the other hand, building a speech recognizer from audio only and relying solely on unsupervised learning techniques is as of today a widely unsolved challenge.

We argue that this completely zero resource scenario may be of great scientific, but perhaps to a lesser extent, of practical interest. For a large number of languages at least some additional resources exist. If the language has a written form, text data is usually much cheaper to acquire than complete pronunciation dictionaries and fully annotated speech databases. Furthermore, assuming the availability of simple word-level transcriptions for some utterances is realistic since many recordings of speech are acquired by prompting speakers which read out written sentences.

In this work, we propose a system suited to work with underresourced languages having a written form which is reasonably close to its phonetics. It consists of three components: an acoustic unit discovery (AUD), an acoustic unit-to-grapheme (AU2G) converter and a word segmentation (WS) algorithm, see Fig. 1. The AUD component operates in an unsupervised fashion on audio only. It delivers, for each discovered acoustic unit, an HMM-based model. The AU2G converter is trained utterance-wise and without any information about word boundaries in the AU sequence. Only a limited number of unaligned word-level transcriptions are utilized which are written as character sequences for the AU2G training. Once trained the AU2G component converts the discovered AU sequence into a character (letter) sequence. Finally, the WS module segments this character sequence into words. Rather than doing this in a completely unsupervised fashion as we did in earlier work [2], the language model used within the word segmentation is initialized on text data. With this initialization the word segmentation results can be greatly improved.

Since neither the number of acoustic units needed nor the size of the vocabulary are known beforehand we employ Bayesian nonparametric models in the AUD and WS stage. With these techniques the number of discovered acoustic units and words can grow with the data. Further, to avoid premature decision making, the AUD outputs lattices rather than a single best sequence.

After a review of related work in Section 2 we describe the different components of the system and their interaction in more detail in Section 3. Section 4 presents the evaluation of the performance of the individual components as well as the overall system. Experiments are carried out for English on the Wall Street Journal CSR (WSJ) corpus, where the data allow for well-controlled investigation of the different parts, and on Xitsonga as an underresourced language.

2. Related work

Recently, there has been an increased interest in low- or zero resource speech recognition. The works can be grouped in two classes. One strain of approaches is concerned with bootstrapping an ASR system from well-trained systems from high-resource languages and a limited amount of labeled training data from the target language. They employ multilingual (e.g., bottleneck) features or acoustic units and adapt the acoustic and language models to the target language [3].

An alternative class of techniques considers a completely zero resource scenario, where neither the acoustic unit inventory nor transcribed data is available, not to mention a pronunciation lexicon. In this second category one may distinguish between “flat” and hierarchical approaches. In the first, word- or phrase-like segments are to be discovered directly from audio [4, 5]. Only few works fall in the second category which follows a hierarchical approach, where acoustic phone-like units are detected at the lower and word-like units as sequences of phone-like units at the higher level of the hierarchy, an approach known to be very challenging [6]. Our earlier work in [7] accounted for

\footnote{1In this contribution we use the words 'character', 'letter' and even 'grapheme' synonymously.}
errors in the first stage by employing a probabilistic pronunciation lexicon. It was, however, restricted to small vocabularies. In [8] the authors describe a system to learn a rather complex inventory of acoustic models at different levels of granularity. Word-like units are obtained by clustering. They perform model estimation by iterative reestimation and unsupervised decoding.

Here we also follow a hierarchical approach. This has the advantage that it can provide “full coverage” transcription, i.e., it can transcribe the whole utterance and not only parts of it, which is in contrast to most approaches in the first category. Further it is not limited to small vocabulary tasks but can be applied in principle to arbitrarily large vocabularies. The key contribution of this paper is that we show a way how to exploit unrelated text data to improve the word discovery from spoken utterances.

Finally, it should be mentioned that many works are concerned with either of the two, i.e., with either unsupervised representation or acoustic unit learning or with unsupervised segmentation or clustering of speech into meaningful units, e.g., [9, 10, 11].

3. System Components

3.1. Acoustic unit discovery

In this work we employ the unsupervised AUD system proposed by Ondel et al. [12]. To tackle the challenge of discovering an unknown number of acoustic units, the AUD is based on a Dirichlet process (DP) Hidden-Markov mixture similar to the model proposed by Lee et al. [13]. However, while the latter use a Gibbs sampler (GS) based on the Chinese Restaurant Process to perform inference, Ondel et al. perform a variational approximation based on the stick breaking construction similar to the approach proposed by Blei and Jordan [14]. They can show that they outperform the GS-based system in terms of accuracy of the discovered units, while reducing the training time by parallel processing at the same time. The model is extensively evaluated on different low-resource tasks in [15].

The temporal structure of acoustic unit sequences is modeled by a left-to-right Hidden-Markov model (HMM) for every acoustic unit and connecting all these HMMs in a phone loop. Since the DP is memoryless, the model contains no temporal dependencies over AU boundaries.

The AUD produces decoding results in the form of AU lattices which are then forwarded to the next processing stage.

3.2. Acoustic unit-to-grapheme conversion

In order to take advantage of a LM trained from textual data, the AU lattices need to be translated into a letter-based format. This can be done by a Phoneme-to-Grapheme converter, such as Sequitur [16] or Phonetisaurus [17]. In this work, the Sequitur converter is employed. It is based on an underlying hidden sequence of joint symbols called graphones. In our case, a graphone consists of either zero or one AU and zero or one letter, forming an AU-letter pair. Thus an AU can be mapped to zero or one letter, and, vice versa, a letter to zero or one AU. Furthermore, a Graphone-LM based on Kneser-Ney smoothing is used to model graphone sequence likelihoods. The model is trained by jointly presenting the character and AU sequences of the training sentences (not of individual words) and performing LM probability estimation by means of an EM algorithm. An advantage of the system is its symmetry, i.e., it can be used for acoustic unit-to-grapheme (AU2G) as well as grapheme-to-acoustic unit (G2AU) conversion.

Since the converter is not suitable for lattice input, we modify the model in the following way:

1. Sequitur is trained on a per-utterance basis by presenting an unaligned word-level transcription and the 1-best label sequence from the acoustic lattice for successively higher LM orders.

2. The highest-order LM is extracted from Sequitur and converted into a Weighted Finite State Transducer.

3. Similar to the approach in [18], two additional transducers are constructed to convert acoustic units and characters into the corresponding graphones.

4. The LM transducer and the conversion transducers are composed to get a AU2G transducer which is able to perform a probabilistic conversion on acoustic lattices.

5. For each utterance in the final test set to be segmented into words, the acoustic lattice is composed with the AU2G transducer to obtain a grapheme (i.e., letter) lattice.

An obvious shortcoming of this approach is the mismatch between training and actual usage: The AU2G system is used on AU lattices while the Sequitur training only sees the 1-best sequences. Furthermore and in contrast to the AUD and the word segmentation, the AU2G system is not based on a full Bayesian formulation of the problem. Both shortcomings could be addressed by making use of the WFST-based system recently proposed by Hannemann et al. [18], but this is left to future work.

3.3. Word segmentation

The word segmentation task is carried out by the system proposed by Heymann et al. [19, 2]. This system is an extension of the one proposed by Neubig et al. [20], which includes a proper treatment of higher-order language models.

The goal of the WS is to identify word boundaries in the sequence of given AUs. The word segmentation relies on the fundamental assumption that the variation of a symbol sequence representing a sentence is higher at word boundaries than it is inside words. Thus, given a sufficiently large corpus of sentences, frequently appearing subsequences will be detected as
forming words. By carrying out multiple iterations over the corpus and jointly segmenting sequences and learning new words and their probabilities, the word segmentation system is able to improve the segmentation over time.

The segmentation is done by applying a lexicon that translates known character sequences into word symbols while unknown sequences are passed through. Then, the resulting possible sequences are weighted with an n-gram LM that explicitly incorporates the word boundaries by a word-end symbol. Since new words have to be learned on the fly, the LM must be able to assign weights to unknown words. Thus, a core component of the system is the nested hierarchical Pitman-Yor language model (NHPYLM) introduced by Mochihashi et al. [21] which is based on the hierarchical Pitman-Yor language model (HPYLM). The close relationship of the HPYLM to a certain kind of Kneser-Ney LM has been shown by Teh [22] and it can be understood as a special way of interpolation and backing off.

For the HPYLM, the zerogram word probability is assumed to be uniform. This probability, however, cannot be computed if the number of words is unknown. Therefore the NHPYLM models the zerogram probabilities by a second character-level HPYLM that weights character sequences ending with a word end symbol. Newly learned words are added to the lexicon whenever hypothesized.

After initializing the language model with text data, the segmentation is thus carried out as follows:

For every iteration:

1. for every sentence in the corpus:
   a. Hypothesize word ends at every possible position, i.e., after every letter.
   b. Translate and weigh all possible sequences with lexicon and LM.
   c. Sample a segmentation from all paths through the weighted sequences.
   d. Parse segmentation: Add new words to lexicon and update LM counts.

2. Resample the hyper parameters of the NHPYLM for every level of the LM

The sampling of a segmentation is performed by forward filtering – backward sampling as described in [21] and is carried out as a blocked Gibbs sampling scheme.

Furthermore, the implementation relies on weighted finite state transducers (WFSTs): The input sequences, the lexicon and the LM are represented by WFSTs. Thus, the translation and weighting steps can be performed by composition. This allows to extend the system to lattice input instead of single character sequences as described by [2]: Here, for every utterance, the lattice is weighted with a letter-only LM first and the best sequence is extracted. Then, a sequence segmentation as above is performed.

If a text corpus for the target language is available, the lexicon and all LMs can be initialized upfront. This is performed by putting a word end token at every word boundary and parsing the corpus in the same manner as done during the iterations. However, this is only possible if the input consists of letter lattices, i.e., after the AU lattices have been passed through a AU2G as explained above. The core assumption is that spoken and written language are sufficiently close and the LM probabilities are thus meaningful to describe the spoken language.

4. Experiments

Experiments were performed on the Wall Street Journal CSR (WSJ) corpus [23] and on the Xitsonga dataset of the 2015 Zero Resource Speech Challenge [24, 25]. We first describe data preparation and experiments on WSJ and then move on to the Xitsonga experiments.

4.1. Data preparation on WSJ

We divided the WSJ corpus in such a way as to mimic a scenario we consider typical for an underresourced language, as described in the introduction: The data sets are built by taking 10783 unique utterances from the WSJ st1284 set and splitting them in two halves, a training set comprising 5392 utterances and a test set with 5391 utterances. The training set is used for the training of the AU2G component as described earlier, while performance evaluation is conducted on the test set. For all setups, a 6-gram grapheme LM is used. After training the AU2G component, the character error rate (AU2G CER) on the test set can be obtained as intermediate quality measure for the AU2G component: The AU2G translation of the 1-best sequence is compared to the reference letter sequence and the total number of substitutions, insertions and deletions is set in relation to the total length of the reference sequences.

The LM of the word segmentation module is initialized by randomly taking up to 10% of the 1,631,456 sentences of the WSJ language model training corpus. The WS stage is evaluated by comparing the resulting segmented strings with the correctly segmented reference transcriptions. The quality is measured with the segmentation F-score and the word error rate (WER) with respect to the reference transcription.

Table 1 shows the results for different experimental setups and different sizes of the text corpus for LM initialization, ranging from 10%, 1%, 0.1%, 0.01% to 0% of the 1,631,456 sentences of the WSJ language model training text corpus. Here, 0% means that the WS language model is learned from scratch, i.e., without the use of text data to learn an initial model. The different setups are explained in the next sections.

4.2. Performance on phoneme lattices

The first set of experiments called WSJ Kaldi aims to assess the impact of the initialization without the additional uncertainty introduced by the unsupervised AU discovery. Therefore, phoneme lattices are produced with a phoneme recognizer trained in a supervised way using the Kaldi toolkit [26]. The recognizer is trained with the complete Kaldi WSJ recipe on the above described training set.

After finishing the acoustic model training, phoneme lattices are created for the train and test set by replacing the usual word-level LM with a bigram phoneme LM in the decoding step. The recognizer results in a phoneme error rate (PER) of 12.72% on the test set.

The AU2G component is trained on the training set with 1-best sequences from the phoneme lattices. The AU2G CER ranges from 21.5% for a bigram grapheme LM to 7.7% for the 6-gram LM, which reflects the low uncertainty in the phoneme recognition stage. The AU2G conversion is performed on the phoneme lattices to obtain character lattices and word segmentation is carried out on the latter.

It can be seen that the word segmentation strongly benefits from LM initialization. For example, the performance measure WER reduces from 61.3% without LM initialization to 24.9% with LM initialization on 10% of the LM data.
4.3. Word segmentation on acoustic unit discovery

In the second set of experiments called WSJ AUD the phoneme recognizer is replaced by the AUD, which discovers AUs in an unsupervised fashion. AU2G conversion and subsequent word segmentation are thus performed on lattices generated by the AUD component.

The unsupervised AUD is learned on the test set. Thus, AUD and WS see the same set for discovery. The AUD discovers 80 different acoustic units, which is about twice as many as there are phonemes in the WSJ lexicon.

To assess the quality of the AUD, two measures are taken into account: Firstly, the normalized mutual information (NMI), which is a measure of the similarity of the discovered acoustic units with the true phoneme sequences (see [15] for details). Secondly, an equivalent unsupervised phoneme error rate (EPER) is computed by mapping each acoustic unit to the best-matching phoneme based on a confusion matrix, adding up the edit distances between the mapped AU sequences and the corresponding reference transcriptions and normalizing on the total length of the reference transcriptions (see [5] for an equivalent unsupervised word error rate).

For the AUD result on WSJ, the NMI of the training set is at 35.9% while the EPER is at 75.2%.

The proposed system emerged as part of the 2016 Jelinek Memorial Summer Workshop on Speech and Language Technologies, which was supported by Johns Hopkins University via DARPA LORELEI Contract No HR0011-15-2-0027, and gifts from Microsoft, Amazon, Google, and Facebook.

4.4. Word segmentation on acoustic unit discovery for Xitsonga

Finally, for the set of experiments called Tso AUD, the AUD experiment is repeated on the Xitsonga dataset of the 2015 Zero Resource Speech Challenge [27]. The dataset consists of 4058 utterances and is randomly split into a training set and a test set, containing 2029 utterances each. Since the database is comparatively small, the AUD is performed on both sets, resulting in 89 acoustic units, an NMI of 44.9% and an EPER of 58.3%.

The AU2G component is trained on the respective training data. This exemplifies an important use case where the AU2G component can be used to enlarge or replace the WS LM initialization corpus. In this work, this was done for Xitsonga but not for WSJ. For WSJ we instead opted for a better isolation of the separate components, by using separate WS LM initialization data. This exemplifies an important use case where the AU2G and WS components are trained independently, e.g. at different sites.

4.5. Importance of word segmentation

One might argue that with the given text data for the LM initialization, there is no need for a word segmentation module, because one could compile a word list already from the text data. This, indeed, would be true, if the text data for LM initialization contained all words that occur in the test set. The WSJ test set contains 10284 unique words, and Table 2 shows which percentage of these words is present in the LM training data. It can be seen that only some fraction of words is known for the LM corpus of size 0.1% and 0.01%, demonstrating that the WS module is indeed an important component. Likewise, the reference transcripts of the Xitsonga test set contain 15090 unique words, while even the 100% fraction of the training set transcriptions contains only 57.3% of these words.

If text data is scarce, the transcriptions of the training corpus can be used to enlarge or replace the WS LM initialization corpus. In this work, this was done for Xitsonga but not for WSJ. For WSJ we instead opted for a better isolation of the separate components, by using separate WS LM initialization data. This exemplifies an important use case where the AU2G and WS components are trained independently, e.g. at different sites.

5. Conclusions

The work reported here was supported by Deutsche Forschungsgemeinschaft (DFG) under contract no. Ha3455/12-1 within the priority program 1527 Autonomous Learning.

The proposed system emerged as part of the 2016 Jelinek Memorial Summer Workshop on Speech and Language Technologies, which was supported by Johns Hopkins University via DARPA LORELEI Contract No HR0011-15-2-0027, and gifts from Microsoft, Amazon, Google, and Facebook.
7. References


