INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

CNN-based phone segmentation experiments in a less-represented language

Céline Manenti, Thomas Pellegrini, Julien Pinquier

Université de Toulouse, UPS, IRIT, Toulouse, France
{celine.manenti, thomas.pellegrini, julien.pinquier}@irit.fr

Abstract

These last years, there has been a regain of interest in unsupervised sub-lexical and lexical unit discovery. Speech segmentation into phone-like units may be a first interesting step for such a task. In this article, we report speech segmentation experiments in Xitsonga, a less-represented language spoken in South Africa. We chose to use convolutional neural networks (CNN) with FBANK static coefficients as input. The models take binary decisions whether a boundary is present or not at each signal sliding frame. We compare the use of a model trained exclusively on Xitsonga data to the use of a bootstrap model trained on a larger corpus of another language, the BUCKEYE U.S. English corpus. Using a two-convolution-layer model, a 79% F-measure was obtained on BUCKEYE, with a 20 ms error tolerance. This performance is equal to the human inter-annotator agreement rate. We then used this bootstrap model to segment Xitsonga data and compared the results when adapting it with 1 to 20 minutes of Xitsonga data.

Index Terms: Convolutional Neural Networks, phonemes, segmentation, under-resourced languages

1. Introduction

Speech segmentation is the process, human (cognitive) or automatic (when performed by a machine), which aims to identify the boundaries between units (words, syllables and phonemes) in a registration or a voice stream. In automatic speech processing is a subproblem that has various applications in automatic speech recognition (ASR). Currently, the automatic segments discovery to identify words or sub-lexical units was driven by interest in unsupervised learning of these units, to build a pronunciation lexicon by identifying words and phones inventory without linguistic knowledge *a priori* [1] or to make connections with the human and language acquisition, particularly by children [2].

In this context, we can mention the growing interest of the scientific community for the automatic processing of languages called little-feature, with the organization of conferences and special sessions dedicated to this theme each year, such as the Workshop on speech technologies for low-resourced languages *SLTU*. To these are added challenges such as *Zero Resource Speech Challenge* [3], which was to identify words or pseudo-words and sub-word units from recordings only. The data used in this challenge were the spontaneous speech corpus BUCKEYE, American English, and also a small corpus of a poorly endowed language, Xitsonga, a language of South Africa.

Deep neural networks (DNN) became popular in signal processing because of their excellent performances, especially in ASR. According to the considered problem, they give similar or better results than GMM. For example, [4] get an absolute gain of 3% in classification of vowels. Neural networks have the characteristic of being adaptable to data and the requested task, approaching the form most suited to the problem. In [5], networks were found to be able to mimic representations close to the filter banks directly when taking time series as input signal.

In this work, we addressed the automatic segmentation into phones by modeling the segment boundaries rather than the segments themselves, in a supervised fashion. We use convolutional neural networks (CNN) as they were shown to outperform DNN for a variety of ASR tasks [6]. After a brief description of our system in Section 2, corpus and assessment metrics in Section 3, we compare different configurations of models (number of neurons, filters), and illustrate the influence of the data used (small quantities, different language) when training neural networks. We also test the use of a network trained for English segmentation on Xitsonga.

2. System description

Figure 1 represents the basic processing pipeline of our segmentation system of speech into phone-like units. The three steps are detailed in the following subsections.

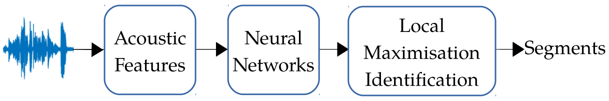

Figure 1: Diagram of the segmentation system

2.1. Acoustic features

Following various tests on time and frequency features, we use 32 filter bank coefficients (FBANK), computed on 16 ms sliding windows with a 4 ms hop size for better precision. We extract the FBANK coefficients and give then as input to the neural network. We recall that the process of extracting FBANK is based on the transformation of the spectral amplitude through a filter bank characterized by triangular filters, linearly distributed along the Mel scale. FBANK coefficients are the energy logarithm of each filter.

2.2. Neural networks

CNN are very efficient in recognizing patterns: for example, more than 99% of correct recognition on handwritten numbers (MNIST) [7]. MLP can achieve similar results, but with more layers: 12 layers fully connected against 6 (1 layer of convolution and 5 fully connected) for a CNN [8]. So, we chose to use a CNN, after having tested MLP and DNN.

For the neural network, the segmentation task is a binary classification task: presence or absence of boundary. Conven-

http://dx.doi.org/10.21437/Interspeech.2016-796

tionally, when assigning a class to a given window, it first calculates for each class the probability that the window belongs to it, then it indicates as output the most probable class. However, this last step presents two difficulties: the two classes (presence, absence of boundary) being divided into unequal proportions (i.e. 1/5, 4/5), the output probabilities favor the absence of boundary. In addition, when a window has a high probability of being a boundary, its neighboring windows are likely to belong to the same class. To avoid this, we post-process the probabilities outputted by the classifier and based the final decision on a method of local maxima identification.

2.3. Local maxima identification

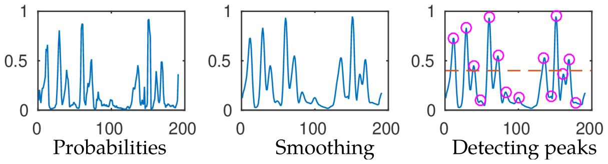

Figure 2: Illustration of our local maxima identification on a 200-sample analysis windows

Figure 2 illustrates the process of finding local maxima. For each analysis window, the neural network calculates a probability that the window contains a boundary. Each recording results in a probability curve, of length 200 samples in Figure 2, for example. To avoid detecting local variations due to noise, we smooth the curve using a convolution with a small Hamming window of size 5 samples. In order to select the more important peaks (local maxima), we only keep those above a threshold. The threshold value can vary as needed to favor precision, recall and F-measure. After few empirical tests, it appeared that the F-measure is maximized when the number of peaks detected is close to that expected, that is on average 1 phone every 70 ms for conversational English or 1 phone every 90 ms for read speech in Xitsonga.

3. Corpora and assessment metrics

We used the American English corpus called BUCKEYE [9], composed of spontaneous speech (radio recordings) collected from 40 different speakers with about 30 minutes of time speech per speaker. This corpus is described in detail in [10]. The quality of the manual phone-level transcriptions was assessed by the creators of the corpus. An inter-annotator agreement was reported: about 76% of correct labels and 62% in F-measure for the manual segmentation with a 10 ms margin (tolerance). The percentage rises to 79% for a tolerance of 20 ms [11]. The median duration of phonemes is about 70 ms, with 60 different phonemes annotated, exceeding the number of 40 phones usually reported for English, especially because of peculiar pronunciations that the creators of BUCKEYE chose to distinguish in different classes, particularly for nasal sounds. Basing ourselves on the cutting of the challenge *Zero Resource Speech*, we divided the whole training sub-corpus in two parts: a training sub-corpus BUCKEYE-TRAIN (75%, 10 hours, 20 speakers), a development corpus BUCKEYE-DEV (25%, 3 hours, 6 speakers), and we kept the official test portion BUCKEYE-TEST (5 hours, 12 speakers) as is.

We performed our segmentation experiments also on a less-resourced language called Xitsonga, a language spoken in South Africa. The Xitsonga corpus [12] is composed of short read sentences recorded on smart-phones, outdoors. We used nearly 500 phrases, with a total of 10,000 examples of phonemes annotated manually, from the same challenge database than the one used in the "*Zero Resource Speech*" challenge. The median duration of phones is about 90 ms and there are 49 different phones. We divided this corpus in a training corpus (Xitsonga-train) of 20 minutes and a testing corpus (Xitsonga-test) of 10 minutes.

4. Experiments

4.1. Comparison of different configurations on BUCKEYE-DEV

In the context of this article, we used Theano [13] and Lasagne [14] for the implementation of the models. Using the input filter bank and with the learning hyper-parameters properly chosen (learning rate = 0.007, regularization coefficient = 0.9, minibatchs of size 2000), CNN proved relevant. So we tried to optimize the settings of the network (number of layers and of neurons, size of convolution filters).

CNN seems to be optimal for our task when using between 50 and 400 neurons for the fully connected layers and the number of convolution filters (also called *maps*) had an impact of around 1% or 2%, absolute. For instance, increasing their number from 15 to 120 filters brings a 1.2% absolute gain. To explain this experimental result, we have analyzed the different filters and we were able to verify that only 15 were active filters for a size of 3x2.

The number of context windows turned out to be one of the most important parameters: changes of phones are located thank to the context. In our experiments, the results improve significantly with the increasing size of context and we chose an optimal value of 18 neighboring windows (84 ms), which is close to the median duration of phones.

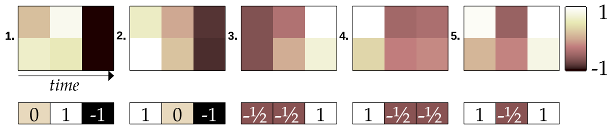

Figure 3: Five examples of CNN filters that seem to perform derivation

The derivative and the second derivative of the FBANK features are often used in ASR systems as input parameters but, for our task, they did not provide any additional information to the CNN compared to using static features only. To understand why, we studied the filters of the first convolution layer. Figure 3 shows five examples of filters with a 3x2 size, with the x-axis being the time axis. We can clearly see that these filters approximate a derivation computation. Studying each of these filters, we see that they approximately perform the following calculations, for a time t and a signal s:

1. $s(t) - s(t+1)$
2. $s(t-1) - s(t+1)$
3. $-\frac{1}{2}s(t-1) - \frac{1}{2}s(t) + s(t+1)$
4. $s(t-1) - \frac{1}{2}s(t) - \frac{1}{2}s(t+1)$
5. $s(t-1) - \frac{1}{2}s(t) + s(t+1)$.

The different approximations of these derivatives are learned by the model directly from the input data, and there-

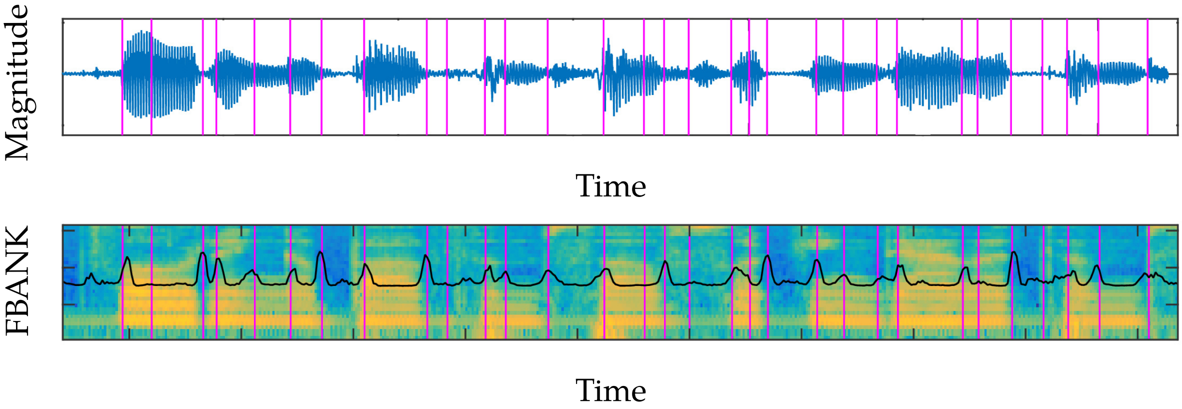

Figure 4: Probabilities output CNN for segmentation on BUCKEYE-TEST. **Top**: temporal signal in blue with manually annotated boundaries represented by vertical bars in purple. **Bottom**: spectrogram of the signal with manually annotated boundaries in purple and CNN probabilities outputs represented by the black curve.

fore, static features seem to be the most appropriate input for our task.

4.2. Results on BUCKEYE-TEST

The dimension of the input layer of our CNN is 18×32, since we use a 18-window context, and 32 FBANK coefficients per window. The CNN is composed of two convolution layers with 3x2 filters (3 for time) and 2x2 filters with 40 filters. Each convolution layer is followed by 2×2 max-pooling layer. Follows a fully connected layer of 200 units before the 2-d output last layer that gives the probability of having a boundary. With this CNN, we obtained F-measures of 68% and 79% with a 10 ms or 20 ms tolerance values, respectively. With a high threshold, we can achieve an accuracy greater than 90% if we agree to find only a sixth of the boundaries. Or, with a very low threshold, we obtain a recall of 72% with half of erroneous detections (see Table 1).

Table 1: BUCKEYE-TEST results for 3 threshold values and 10 ms of tolerance

Phone median size (ms)	Precision	Recall	F-measure
52	0.52	**0.72**	0.61
72	0.71	0.65	**0.68**
272	**0.94**	0.16	0.27

Figure 4 is an example of a result obtained by the neural network, showing the curve of probabilities superimposed to the signal spectrogram. The high values of the curve actually correspond to changes in the spectrum and are correlated with the boundaries.

We analyzed the boundary detection rate of some phones among the most frequent ones. We find that boundaries of phones with a strong attack, such as [g] or [k] , are more easily found that for [l] or [ɹ], who encounter more difficulties. We also observed that boundaries between two consecutive vowels are difficult to retrieve, especially because is a slow and small variation. In addition, annotators noticed that precision of boundary depends of the size of the phone. For example, boundaries between [oʊ] and [aɪ] are rarely found.

Globally speaking, the results are close to the inter-annotator agreement between human annotators. Table 2 even shows that our system is more accurate when it locates a true boundary: we have a better F-measure for 10 ms of error tolerance and its increase between 10 ms and 20 ms is lower than that observed between annotators. In comparison, for an error tolerance of 20 ms, a random baseline is around 47%. For information, a state-of-the-art result reported in [15] reaches 77% in F-measure on another corpus: TIMIT, with 20 ms of tolerance.

Table 2: Comparison of F-measures between human annotators and the CNN on BUCKEYE-TEST

Tolerance (ms)	Random	Annotators	CNN
10	0.26	0.62	0.68
20 ms	0.47	0.79	0.79

4.3. Application to a poorly endowed language (Segmentation with little training data)

Segmentation is a ”simple” binary task. So we can expect that little data would be sufficient to train a model, or that using a model learned with a different language for which large datasets are available could help detecting boundaries in a less-resourced language.

From BUCKEYE to the Xitsonga corpus, the only parameter to adjust is the threshold applied to the output of the network. Choosing the threshold value is equivalent to choosing the median size of inferred segments. As can be seen in the Figure 5, the optimal phone duration that maximizes the F-measure is close to the data-driven one.

Using our model trained on U.S. English, we obtained a 62% F-measure with a 20 ms error margin on the Xitsonga test data. Performance was much lower with a 10 ms error margin. To better understand this decrease, we measured the evolution of the F-measure depending on the amount of BUCKEYE training data. We observed that during training on BUCKEYE,

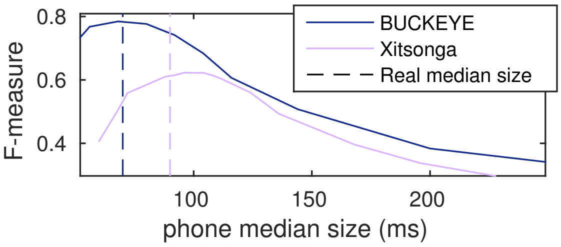

Figure 5: Influence on the F-measure of the segment median size, segments obtained with different thresholds. For reference, the real median size of the phones tested corpus is indicated by dotted lines. Results for a network learned with BUCKEYE-TRAIN only.

performance on BUCKEYE-TEST kept increasing when more training epochs were performed, while performance quickly reached a plateau when testing the model on the Xitsonga-test data. In contrast, with a 20 ms tolerance, performance on both English and Xitsonga data kept increasing. So it seems that training on a different language helps locating boundaries approximately, with a lack of precision in time. In order to increase precision in time, we tried to adapt the English model with Xitsonga data.

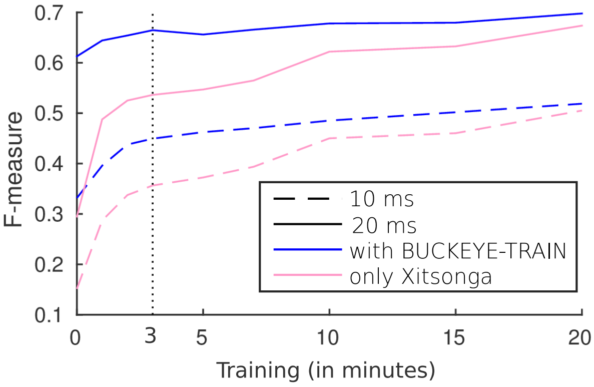

Figure 6: Increase in F-measure based on the number of minutes of Xitsonga for training.

Figure 6 represents the F-measure values obtained on the Xitsonga-test corpus as a function of the number of minutes of Xitsonga speech data used to adapt the bootstrap English model (plain and dotted lines in blue) or used to train a model from scratch (plain and dotted lines in red). For a margin of 20 ms (plain lines), the bootstrap model outperforms the model trained on Xitsonga data only. Adding only three minutes of Xitsonga training data achieves a 65% of F-measure, a value close to the best performance obtained with 20 minutes.

Adding a few minutes of training data greatly improved these results, as shown in the figure. Indeed, only three minutes brought about 10% absolute improvement in F-measure for a 10 ms margin. But using only these three minutes of Xitsonga allows to match the results achieved by the bootstrap English model for the same margin. The baseline English model visibly seems to be more useful with a larger margin of error (20 ms). There are several possible ways to adapt the bootstrap model. Transfer learning, or more simply model adaptation, is usually performed by retraining the deepest layers of a network. In our case, simply retraining the output layer did not provide any improvement. Improvements only occurred when retraining at least the last dense hidden layer. Furthermore, retraining all the layers was the best option according to the tests we did.

Still studying figure 6, we see that adapting with 20 minutes of Xitsonga outperforms from about 2% absolute the model trained from scratch. Thanks to this Xitsonga-train corpus, we get 52% of F-measure for 10 ms and 70% for 20 ms, that is respectively 16% and 9% less than BUCKEYE-TEST. The curves show that convergence is not finished and that more data could improve the results. Another parameter to consider is the mean duration of the phones. Indeed, the mean duration of the Xitsonga phones being superior by a $2/7$ factor compared to the mean duration of the English phones from BUCKEYE, we can assume that it penalizes the results on Xitsonga for a given margin of error.

Probably related to the difference in mean duration of their phones, we noticed that the probabilities of the boundary predictions were lower for the network trained on Xitsonga only, compared to the bootstrap model. We can perhaps explain this observation because of the smaller ratio of the number of boundaries/non-boundaries in the Xitsonga corpus than in BUCKEYE, due to longer phones in Xitsonga. The network therefore sees a lower proportion of boundaries and tends to assign lower probabilities to the boundary class. We also found that the two networks detect approximately the same boundaries. Despite the difference in scale on the borders, the probabilities of the two models have a 0.91 correlation rate.

5. Conclusions

In this article, we reported speech segmentation experiments at phone-level using CNN. We envisaged the segmentation task as a binary classification problem in which the classifier has to decide on the presence of an eventual phone boundary based on FBANK coefficients as input. On the American English recordings of the BUCKEYE corpus, a CNN with two convolution layers achieved some remarkable results: 68% of F-measure for our best automatic segmentation system versus 62% for the inter-annotator agreement with a tolerance of 10 ms on the location of phones boundaries. Moreover, the models showed good adaptation to difficult cases: little training data and application to another less-resourced language, Xitsonga, a language spoken in South Africa. We used a CNN trained on BUCKEYE as a bootstrap to segment Xitsonga speech data. This model achieved a 62% F-measure with a 20 ms error margin on this data, which shows that using a model trained on a given language can be used to segment speech from another language. Then, we adapted the bootstrap model with little Xitsonga data. Using only 3 minutes of adapting data brought a 10% F-measure improvement with a 10 ms error tolerance. We plan to confirm our findings by using larger quantities of adapting data and also by testing the bootstrap models on other languages. For the segmentation of under-resourced languages, such as Xitsonga, we also plan to explore semi-supervised learning, i.e. segmenting unseen data and then re-using this data as training data. Another direction would be to investigate better feature representations with techniques such as auto-encoders.

6. References

[1] C.-y. Lee, T. J. O'Donnell, and J. Glass, "Unsupervised lexicon discovery from acoustic input," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 389–403, 2015.

[2] A. Jansen, E. Dupoux, S. Goldwater, M. Johnson, S. Khudanpur, K. Church, N. Feldman, H. Hermansky, F. Metze, R. C. Rose, M. Seltzer, P. Clark, I. McGraw, B. Varadarajan, E. Bennett, B. Brschinger, J. Chiu, E. Dunbar, A. Fourtassi, D. Harwath, C. ying Lee, K. Levin, A. Norouzian, V. Peddinti, R. Richardson, T. Schatz, and S. Thomas, "A summary of the 2012 jhu clsp workshop on zero resource speech technologies and models of early language acquisition." in *INTERSPEECH*, 2013, pp. 8111–8115.

[3] M. Versteegh, R. Thiollire, T. Schatz, X. N. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015," in *INTERSPEECH*, 2015, pp. 3169–3173.

[4] S. Joshi, N. Deo, and P. Rao, "Vowel mispronunciation detection using dnn acoustic models with cross-lingual training." in *INTERSPEECH*, 2015, pp. 697–701.

[5] M. Bhargava and R. Rose, "Architectures for deep neural network based acoustic models defined over windowed speech waveforms." in *INTERSPEECH*, 2015, pp. 6–10.

[6] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 8614–8618.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *INTERSPEECH*, 1998, pp. 2278–2324.

[8] P. Golik, Z. Tske, R. Schlter, and H. Ney, "Convolutional neural networks for acoustic modeling of raw time signal in lvcsr," in *INTERSPEECH*, 2015, pp. 26–30.

[9] M. Pitt, L. Dilley, K. Johnson, S. Kiesling, W. Raymond, E. Hume, and E. Fosler-Lussier, "Buckeye corpus of conversational speech (2nd release)," www.buckeyecorpus.osu.edu, 2007.

[10] S. Kiesling, L. Dilley, and W. D. Raymond, "The variation in conversation (vic) project: Creation of the buckeye corpus of conversational speech," in *Language Variation and Change*, 2006, pp. 55–97.

[11] W. D. Raymond, M. Pitt, K. Johnson, E. Hume, M. Makashay, R. Dautricourt, and C. Hilts, "An analysis of transcription consistency in spontaneous speech from the buckeye corpus," in *ICSLP*, 2002.

[12] C. van Heerden, M. Davel, and E. Barnard, "The semi-automated creation of stratified speech corpora," 2013.

[13] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[14] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. D. Fauw, M. Heilman, diogo149, B. McFee, H. Weideman, takacsg84, peterderivaz, Jon, instagibbs, D. K. Rasul, CongLiu, Britefury, and J. Degrave, "Lasagne: First release." Aug. 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.27878

[15] C. Lee, T. O'Donnell, and J. Glass, "Unsupervised lexicon discovery from acoustic input," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 389–403, 2015.