Interspeech 2018
2-6 September 2018, Hyderabad

Task specific sentence embeddings for ASR error detection

Sahar Ghannay, Yannick Estève, Nathalie Camelin

LIUM - University of Le Mans, France
firstname.lastname@univ-lemans.fr

Abstract

This paper presents a study on the modeling of automatic speech recognition errors at the sentence level. We aim in this study to compensate certain phenomena highlighted by the analysis of outputs generated by an ASR error detection system we previously proposed. We investigated three different approaches, that are based respectively on the use of sentence embeddings dedicated to ASR error detection task, on a probabilistic contextual model, and on a bidirectional long short-term memory (BLSTM) architecture. An approach to build task-specific sentence embeddings is proposed and compared to the Doc2vec approach. Experiments are performed on transcriptions generated by the LIUM ASR system applied to the French ETAPE corpus. They show that the proposed sentence embeddings dedicated to ASR error detection achieve better results than generic sentence embeddings, and that the integration of task-specific embeddings in our system achieves better results than the probabilistic contextual model and BLSTM models.
Index Terms: Error detection, Speech recognition, Neural networks, Sentence embeddings.

1. Introduction

Automatic Speech Recognition (ASR) systems continue making errors during speech processing, especially when handling various phenomena, including *e.g.* acoustic conditions (noise, competing speakers, channel conditions), out of vocabulary words, pronunciation variations, *etc.* These errors can have a considerable impact on applications based on the use of automatic transcriptions, like speech to speech translation, spoken language understanding, *etc.*

Error detection aims to improve the exploitation of ASR outputs by downstream applications. For two decades, many studies have focused on the ASR error detection task. Usually, the best ASR error detection systems were based on the use of Conditional Random Fields (CRF) [1]. In [2], the authors detect error regions generated by Out Of Vocabulary (OOV) words. They propose an approach based on a CRF tagger, which takes into account contextual information from neighboring regions instead of considering only the local region of OOV words. A similar approach for other kinds of ASR errors is presented in [3]: the authors propose an error detection system based on a CRF tagger using various ASR-derived, lexical and syntactic features. Recent approaches leverage neural network classifiers. In [4, 5], the authors investigated three types of ASR error detection tasks, *e.g.* confidence estimation, out-of-vocabulary word detection and error type classification, based on a deep bidirectional recurrent neural networks. In our previous studies [6, 7, 8], we investigated the use of several types of continuous word representations. In [6], we proposed a neural approach to detect errors in automatic transcriptions, and to calibrate confidence measures provided by an ASR system. In addition, we studied different word embeddings combination approaches in order to take benefit from their complementarity. We proposed as well to enrich our ASR error detection system with acoustic information which is obtained through acoustic embeddings [7, 8].

In this paper, we propose first to recall the newest results obtained by our system, that combines prosodic features and acoustic embeddings in addition to the other features. Then, we present the main contribution of this study, that consists in modeling automatic speech recognition errors at the sentence level. This study aims to compensate certain phenomena highlighted by the analysis of the outputs generated by the ASR error detection system we previously proposed. We investigated three different approaches, that are based respectively on the use of sentence embeddings dedicated to ASR error detection task, a probabilistic contextual model and a bidirectional long short term memory (BLSTM) architecture. An approach to build task-specific sentence embeddings is proposed and compared to the Doc2vec [9] approach.

2. ASR error detection

An ASR error detection system attributes a label *Error* or *Correct* for each word in the automatic transcription, by analyzing each word within its context. This analysis is based on a set of features defined below. The context window size used in this study is 2 on both sides of the current word.

The proposed neural architecture is a feed forward neural network, based on a multi-stream strategy to train the network, named MultiLayer Perceptron MultiStream (MLP-MS). A detailed description of this architecture was presented in a previous study [10].

Each word is represented by a feature vector composed of the following features: **ASR features** are the posterior probabilities generated from the ASR system at the word level. **Lexical features** are the length of the current word and three binary features indicating if the three 3-grams containing the current word have been seen in the training corpus of the ASR language model. **Syntactic features** are POS tag, dependency labels and word governors, which are extracted using the MACAON NLP Tool chain[1] [11] to process the ASR transcriptions. **Prosodic features** are number of phonemes, average duration of phonemes, average f0 of the word, *etc.*, those features are detailed in a previous study [6]. In addition to those features, **linguistic and acoustic embeddings** are used. The **linguistic embeddings** correspond to the combination through an auto-encoder of *word2vecf* based on dependency trees [12], *skip-gram* provided by *word2vec* [13], and *GloVe* [14]. The *acoustic embeddings* are built using the approach proposed by [15], which allows to build acoustic signal embeddings of words observed in an audio corpus, and also acoustic word embeddings of words never observed in this corpus, by exploiting their orthographic representations.

[1] http://macaon.lif.univ-mrs.fr

10.21437/Interspeech.2018-2211

3. Experiments

3.1. Experimental setup

Experimental data for ASR error detection is based on the entire official ETAPE corpus [16], composed by audio recordings of French broadcast news shows, with manual transcriptions (reference). This corpus is enriched with automatic transcriptions generated by the LIUM ASR system, that won the ETAPE evaluation campaign in 2012. A detailed description is presented in [17]. The experimental corpus is divided into three sets: Train, Dev and Test, which are composed respectively, of $349K$, $54K$, and $58K$ words. Their word error rates are 25.3%, 24.6% and 21.9% respectively.

The linguistic word embeddings were computed from a large textual corpus in French, composed of about 2 billions of words. This corpus was built from articles of the French newspaper "Le Monde", the French Gigaword corpus, articles provided by Google News, and manual transcriptions of about 400 hours of French broadcast news. While, the acoustic embeddings are trained on 488 hours of French Broadcast News with manual transcriptions. A detailed description of the data and the architectures is given in [7].

3.2. Experimental results

This section reports the experimental results got on the data set using the ASR error detection system. The performance is evaluated by using recall (R), precision (P) and F-measure (F) for the erroneous word prediction and global Classification Error Rate (CER). The CER is defined as the ratio of the number of misclassifications over the number of recognized words. The significant results are underlined and measured using the 95% confidence interval.

These results concern the evaluation of the combination of prosodic features with acoustic embeddings, in addition to other features described in section 2 including the linguistic embeddings. The performance of our new system, denoted as *Sys2*, is compared with the previous one proposed in [7], denoted as *Sys1*. The latter integrates only the acoustic embeddings and the features described in section 2.

Table 1: *Performance of the combination of prosodic features and acoustic embeddings in addition to the other features on Dev and Test corpora.*

Corpus	System	Label *Error*			Global
		P	**R**	**F**	**CER**
Dev	*Sys1*	0.71	0.58	0.64	9.53
	Sys2	0.71	0.60	0.65	**9.38**
Test	*Sys1*	0.70	0.59	0.64	7.94
	Sys2	0.70	0.61	0.65	**7.75**

Experimental results reported in Table 1 show the usefulness of prosodic features when combined to acoustic embeddings. This combination yields an interesting improvement in terms of CER reduction in comparison to the results of *Sys1*.

3.3. Average span analysis of the ASR error detection system outputs

In this section, we are interested in the analysis of the outputs of our best system: *Sys2*, in order to perceive the errors that are hard to detect. This analysis is performed based on the average error segment size (average span), since we know that our system takes only local decisions and is not designed to perform optimally sequence predictions.

Results, summarized in Table 2, present the average span and the standard deviation for the ground truth, the predictions (classifier outputs) and the correct predictions for *Sys2*. The average span of the correct predictions is defined as the average error segment of the contiguous errors correctly detected.

Table 2: *The average span and the standard deviation for the ground truth, the predictions, and the correct predictions for Sys2.*

Corpus		**Average span**	**Standard deviation**
Train	Ground truth	3.03	1.72
Dev		3.24	2.15
Dev	Predictions	2.82	1.28
	Correct predictions	2.66	1.05

We observe that for the *Sys2* system the average span of predictions is smaller by 12.9% compared to the ground truth, with a smaller standard deviation by 40.5%. We also notice that the average span for correct predictions is much smaller than the ground truth. The gap related to the error segment size between the ground truth, the predictions and the correct predictions is due to the architecture of *Sys2* system, since this one takes only local decisions and is not currently designed to perform optimally sequence prediction.

The analysis results provided us useful information in order to improve the performance of the proposed ASR error detection system. For this purpose, we propose to explore the use of global information, at the sentence level, and evaluate their impact by using the same neural architecture.

4. Global decision: sentence embeddings

4.1. Sentence Embeddings

In this section, we focus on integrating global information to enrich our ASR error detection system, through the use of sentence embeddings (Sent-Emb). These representations have been successfully used in sentence classification and sentiment analysis tasks [9, 18, 19]. Sentence embeddings can also be built in a generic context by using the tool Doc2vec [9], or they can be adapted to a specific task like for the sentiment analysis task, as in [20].

For the error detection task, we propose to build sentence embeddings that carry information about the confidence of a recognition hypothesis at the sentence level: whether the sentence is almost correct or highly erroneous. Then, we compare the performance of the proposed sentence embeddings to the DBOW (Distributed bag of words) embeddings provided by Doc2vec [9]. In our experiments, the DBOW model is trained on the ETAPE corpus to build 100-dimensional embeddings, named Emb_{DBOW}, for each automatic transcription (utterance).

4.1.1. Task-specific embeddings

The sentence embeddings Emb_{DBOW} carry semantic information held in automatic transcriptions, but probably do not carry information specific to ASR error detection task. Thus, we propose to build specific sentence embeddings for this task.

For this purpose, we propose to use embeddings extracted from a convolution neural network (CNN), named Emb_{CNN}. The CNN is trained to predict whether an automatic transcription (utterance) is slightly erroneous (SE) or very erroneous (VE). To build those embeddings we need to use a labeled corpus: each utterance in the ETAPE corpus was tagged to "slightly erroneous" or "very erroneous". In this study, we arbitrarily consider a recognition hypothesis as very erroneous if 20% of its words are incorrect. Utterances with less than 20% of incorrect words are then considered as slightly erroneous (including fully correct utterances).

Table 3 presents the description of the data used to train the convolution neural network.

Table 3: *Description of the data used to build the Emb_{CNN} embeddings: number of reference and hypotheses utterances and the number of "SE" and "VE" utterances.*

Coprus	#Ref. Utt.	# Hyp. Utt.	#SE Utt.	#VE Utt.
Train	22K	21.3K	13.3K	8.3K
Dev	3.7K	3.5K	2.2k	1.3k
Test	3.6K	3.5K	2.3K	1.1K

The CNN takes as input an utterance represented by a vector of features and provides as outputs two labels "SE" or "VE". It is composed of two convolution and max pooling layers followed by two fully connected layers. From the hidden layer just before the Softmax layer we extracted the 100-dimensional sentence embeddings (Emb_{CNN}) for each utterance. Note that, the CNN classifier achieves 13.5% of classification error rate on Test corpus transcriptions.

The utterance feature vector corresponds to the concatenation of feature vectors of words (described in section 2) composing the utterance. The size of utterances is set to 50 words, since 98.37% of them have a size that varies between 1 and 50 words. When utterances are shorter, they are padded with zero equally on both ends, while longer utterances are cut equally on both ends.

The figure 1 shows the MLP-MS architecture that integrates all the features. The feature vectors described in section 2 of the current word and its neighbors (w_x), the sentence embeddings (Emb_{DBOW} or Emb_{CNN}) and the acoustic embeddings (s_i and w_i^+) were processed separately by a specific streams.

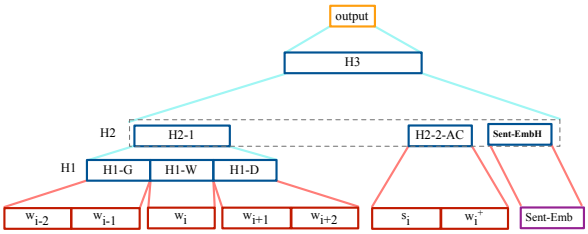

Figure 1: *MLP-MS architecture for ASR error detection task, that integrates acoustic and sentence embeddings in addition to the other features including the linguistic embeddings.*

4.1.2. Experimental results

This section summarizes the comparison results between both sentence embeddings: Emb_{DBOW} and Emb_{CNN}. Performances reached by using those embeddings are compared to the ones got by *Sys2*. Experimental results, summarized in table 4, show that the integration of sentence embeddings was helpful and yields to some improvements in comparison to the results of *Sys2*, especially when using the Emb_{CNN} embedding, which is better than the Emb_{DBOW} embedding. The Emb_{CNN} embedding yields to 1.27% and 0.77% of CER reduction in comparison to *Sys2*, respectively on Dev and Test. This system is named *Sys3* further in the paper. From these results we can reveal that the Emb_{CNN} have captured information about the error useful for our targeted task.

Table 4: *Performance of sentence embeddings Emb_{DBOW} and Emb_{CNN} in comparison to the results of Sys2 system on Dev and Test corpora*

Corpus	Sentence Embed.	Label *Error*			Global
		P	R	F	CER
Dev	- (*Sys2*)	0.72	0.60	0.65	9.38
	Emb_{DBOW}	**0.73**	0.58	0.65	9.36
	Emb_{CNN}	0.72	0.60	0.65	**9.26**
Test	- (*Sys2*)	0.70	0.61	0.65	7.75
	Emb_{DBOW}	**0.72**	0.57	0.64	7.72
	Emb_{CNN}	**0.72**	0.58	0.64	**7.69**

4.2. Probabilistic contextual model

The probabilistic contextual model (PCM) is an other approach we can explore to compensate the phenomena highlighted by analyzing the outputs generated by *Sys2* system. We assume that this approach, that carries information on error distribution, will solve the problem of error segment size, poorly detected by our system.

This approach is similar to the one used in [21] to detect spontaneous speech segments. The authors proposed to extend a local classification process using a probabilistic contextual tag-sequence model that takes into consideration information of surrounding segments in a window of size 3. With this extension, the labeling, which was resulting from a succession of local decisions, becomes a global process.

We propose to apply this idea to our approach to detect ASR errors. We hope to smooth the classification results at the sentence level, by taking into account the local classification of the neighboring words in a window of size 5 similar to the one used in the input of our ASR error detection system. For this reason, we investigated the use of a n order probabilistic model of error distribution: this model estimates the probability that the current word is erroneous according to the accuracy of the 4 neighboring words.

4.2.1. Results

We used the tool *OpenFst*[2] to create the model on the automatic transcripts of the ETAPE corpus and the outputs of two ASR error detection systems: *Sys2* and *Sys3*. The resulting systems are named with extension **PCM**. The results obtained by this approach are summarized in the table 5.

We observe that the application of PCM model to the outputs of *Sys2* yields to slight improvements in terms of CER reduction on both Dev and Test corpora. These results are comparable to the ones obtained by *Sys3*, that integrates the task specific sentence embeddings. However, the application of PCM

[2]http://www.openfst.org/twiki/bin/view/FST/WebHome

Table 5: *Performance of probabilistic contextual model applied to Sys2-PCM and Sys3-PCM systems, on Dev and Test*

Corpus	*System*	**tiquette *erreur***			**Global**
		P	**R**	**F**	**CER**
Dev	*Sys2-PCM*	**0.73**	0.56	0.65	9,31
	Sys3-PCM	0.73	0.60	0.65	**9.23**
Test	*Sys2-PCM*	0.72	0.59	0.65	**7.67**
	Sys3-PCM	0.73	0.57	0.64	7.69

model to the *Sys3* system outputs improved slightly the results only on Dev. This can be explained by the fact that this system already incorporates knowledge about the sentence, and the information provided by the PCM approach is redundant.

4.3. Comparison to bidirectional LSTM system

These last experiments revealed the usefulness of the sentence embeddings integration in our MLP-MS architecture. Since some neural architectures showed recently to be effective to process sequence to sequence tasks [22], it could be interesting to compare them to the neural approach used until now in our experiments. By this way, we want to measure the impact of the use of continuous representations in an MLP architecture to the use of a bidirectional LSTM architecture. A such architecture is designed to learn how to integrate relevant long distant information, and was successfully used for the ASR error detection task in [5, 4].

In our experiments, the bidirectional LSTM architecture is composed of two hidden layers of 512 hidden units each, *i.e.* 256 units in each forward and backward sides. It integrates the same features as the *Sys2* system, without sentence embeddings, we call it BLSTM. Results summarized in table 6 show that *BLSTM* and *Sys2* systems obtain comparable results.

Table 6: *Results on ASR error detection using BLSTM architecture.*

Corpus	*System*	**Label *Error***			
		P	**R**	**F**	**CER**
Dev	BLSTM	0.70	0.63	0.67	9.28
Test	BLSTM	0.69	0.63	0.66	7.83

Notice that *BLSTM* system obtains better results on Dev but not on Test corpus: it seems that the BLSTM architecture did not generalize well in these experiments probably due to a too small size of training data, since this architecture has many parameters to train.

In this paper, we focus on word and sentence continuous representations, and evaluate them for the ASR error detection task through the use of a feedforward neural architecture. These results with the BLSTM architecture, recently proposed for this task, validate our previous experiments, and show that they cannot be questioned in relation to the use of a more sophisticated neural architecture.

Moreover, these results confirm our hypothesis about the integration of global information in our MLP-MS system in order to take better local decisions, since *Sys3* system achieves better results than *BLSTM* system.

4.4. Average span analysis of the ASR error detection system outputs

We revealed that the addition of the information extracted at the sentence level improves the performance of our system. In order to confirm the hypotheses discussed in section 3.3, we report in this section the results of the average span analysis performed on the *Sys3* system outputs in addition to the *BLSTM* ones. Table 7 presents the average spans and the standard deviations for the ground truth, the predictions and the correct predictions for *Sys2*, *Sys3* and *BLSTM* systems. The results show that sentence embeddings have captured information about the error propagation: indeed, the addition of these embeddings (system *Sys3*) has improved the average span compared to the *Sys2* system. We observe as well that the use of *BLSTM* system has improved the average span compared to the *Sys2* system. It achieves competitive results to *Sys3* system.

Table 7: *The average span and the standard deviation for the ground truth, the predictions, and the correct predictions for Sys2, Sys3 and BLSTM systems on Dev.*

Approach		**Average span**	**Standard deviation**
	Ground truth	3.24	2.15
Sys2	Predictions	2.82	1.28
	Correct predictions	2.66	1.05
Sys3	Predictions	3.15	1.70
	Correct predictions	2.84	1.22
BLSTM	Predictions	3.40	2.16
	Correct predictions	2.95	1.40

5. Conclusion

This paper presents a study on modeling the ASR errors at the sentence level to compensate certain phenomena highlighted by the analysis of the outputs generated by the ASR error detection system we previously proposed. We experimented the use of three different approaches, that are based respectively on the use of sentence embeddings dedicated to ASR error detection task, a probabilistic contextual model, and a BLSTM architecture. In addition, we proposed an approach to build task-specific sentence embeddings and compare it to the Doc2vec approach. Experiments, that were performed on the French ETAPE corpus, show the high complementarity of acoustic word embeddings and prosodic information, and show that the proposed task-specific sentence embeddings achieve better results than the general ones proposed by Doc2vec. Moreover, their integration into our system improves the results in comparison to the application of the PCM model on the *Sys2* outputs and also in comparison to the use of a BLSTM.

6. Acknowledgements

This work was supported by the French ANR Agency through the CHIST-ERA M2CR project, under the contract number ANR-15-CHR2-0006-01

7. References

[1] Lafferty, John D. and McCallum, Andrew and Pereira, Fernando C. N., "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, San Francisco, CA, USA, 2001, ICML '01, pp. 282–289, Morgan Kaufmann Publishers Inc.

[2] Carolina Parada, Mark Dredze, Denis Filimonov, and Frederick Jelinek, "Contextual Information Improves OOV Detection in Speech," in *Human Language Technologies: Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL'10)*, 2010.

[3] Frédric Béchet and Benoit Favre, "ASR error segment localization for spoken recovery strategy," in *EEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*, May 2013, pp. 6837–6841.

[4] Atsunori Ogawa and Takaaki Hori, "Error detection and accuracy estimation in automatic speech recognition using deep bidirectional recurrent neural networks," *Speech Communication*, vol. 89, pp. 70–83, 2017.

[5] Atsunori Ogawa and Takaaki Hori, "ASR error detection and recognition rate estimation using deep bidirectional recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference*. IEEE, 2015, pp. 4370–4374.

[6] Sahar Ghannay, Yannick Estève, Nathalie Camelin, Camille Dutrey, Fabian Santiago, and Martine Adda-Decker, "Combining continous word representation and prosodic features for ASR error prediction," in *3rd International Conference on Statistical Language and Speech Processing (SLSP 2015)*, Budapest (Hungary), November 24-26 2015.

[7] Sahar Ghannay, Yannick Estève, Nathalie Camelin, and Paul Deleglise, "Acoustic word embeddings for ASR error detection," in *Interspeech 2016*, San Francisco (CA, USA), 9-12 September 2016.

[8] Sahar Ghannay, Yannick Esteve, Nathalie Camelin, and Paul Deléglise, "Evaluation of acoustic word embeddings," in *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, 2016, pp. 62–66.

[9] Quoc V Le and Tomas Mikolov, "Distributed Representations of Sentences and Documents.," in *ICML*, 2014, vol. 14, pp. 1188–1196.

[10] Sahar Ghannay, Yannick Estève, and Nathalie Camelin, "Word embeddings combination and neural networks for robustness in ASR error detection," in *European Signal Processing Conference (EUSIPCO 2015)*, Nice (France), 31 aug.-4 sept. 2015.

[11] Alexis Nasr, Frédéric Béchet, Jean-François Rey, Benoît Favre, and Joseph Le Roux, "Macaon: An nlp tool suite for processing word lattices," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, 2011, pp. 86–91.

[12] Omer Levy and Yoav Goldberg, "Dependency-based word embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, vol. 2, pp. 302–308.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," in *arXiv preprint arXiv:1301.3781*, 2013.

[14] Jeffrey Pennington, Richard Socher, and Christopher D Manning, "Glove: Global vectors for word representation.," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, vol. 14, pp. 1532–1543.

[15] Samy Bengio and Georg Heigold, "Word embeddings for speech recognition.," in *INTERSPEECH*, 2014, pp. 1053–1057.

[16] Guillaume Gravier, Gilles Adda, Niklas Paulsson, Matthieu Carr, Aude Giraudel, and Olivier Galibert, "The ETAPE corpus for the evaluation of speech-based TV content processing in the French language," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, 2012.

[17] Paul Deléglise, Yannick Estève, Sylvain Meignier, and Teva Merlin, "Improvements to the LIUM French ASR system based on CMU Sphinx: what helps to significantly reduce the word error rate?," in *Interspeech*, Brighton, UK, September 2009.

[18] Yiou Lin, Hang Lei, Jia Wu, and Xiaoyu Li, "An empirical study on sentiment classification of chinese review using word embedding," *arXiv preprint arXiv:1511.01665*, 2015.

[19] Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, and Ming Zhou, "Sentiment embeddings with applications to sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 496–509, 2016.

[20] Yafeng Ren, Ruimin Wang, and Donghong Ji, "A topic-enhanced word embedding for Twitter sentiment classification," *Information Sciences*, vol. 369, pp. 188–198, 2016.

[21] Richard Dufour, Yannick Estève, and Paul Deléglise, "Characterizing and detecting spontaneous speech: Application to speaker role recognition," *Speech Communication*, vol. 56, pp. 1–18, 2014.

[22] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.