WORD EMBEDDINGS COMBINATION AND NEURAL NETWORKS FOR ROBUSTNESS IN ASR ERROR DETECTION

Sahar Ghannay, Yannick Estève, Nathalie Camelin

LIUM - University of Le Mans, France

ABSTRACT

This study focuses on error detection in Automatic Speech Recognition (ASR) output. We propose to build a confidence classifier based on a neural network architecture, which is in charge to attribute a label (error or correct) for each word within an ASR hypothesis. This classifier uses word embeddings as inputs, in addition to ASR confidence-based, lexical and syntactic features. We propose to evaluate the impact of three different kinds of word embeddings on this error detection approach, and we present a solution to combine these three different types of word embeddings in order to take advantage of their complementarity.

In our experiments, different approaches are evaluated on the automatic transcriptions generated by two different ASR systems applied on the ETAPE corpus (French broadcast news). Experimental results show that the proposed neural architectures achieve a CER reduction comprised between 4% and 5.8% in error detection, depending on test dataset, in comparison with a state-of-the-art CRF approach.

Index Terms— Automatic speech recognition, confidence measures, neuronal networks, word embeddings.

1. INTRODUCTION

The advancement in the speech processing field and the availability of powerful computing devices have led to better performance in the speech recognition domain. However, recognition errors are still unavoidable, whatever the quality of the ASR systems. This reflects their sensitivity to the variability: the acoustic environment, speaker, language styles and the theme of the speech. These errors can have a considerable impact on the application of certain automatic processes such as information retrieval, speech to speech translation, etc.

Error detection can help to improve the exploitation of ASR outputs by downstream applications, but is a difficult task given the fact that there are several types of errors, which can range from the simple substitution of a word with a homophone to the insertion of an irrelevant word for the overall understanding of the sequence of words. They can also affect neighboring words and create a whole area of erroneous words.

In this paper, we tackle the problem of ASR error detection. Firstly we investigate the use of word embeddings as input features of the error detection system. We experiment the use of three different types of word embeddings and propose to combine them with an auto-encoder in order to take advantage of their complementary. Secondly, we propose several confidence classifiers that attribute a label (error or correct) for each word of an ASR hypothesis. They are built on neural network architectures, including simple Multi Layer Perceptron (MLP) or MLP Multi Stream, and are compared with approaches recently proposed in the literature. Particularly, we are interested in the re-use of such classifiers for different ASR systems, without re-training these classifiers. This can be useful when ASR outputs of a new system are provided without data enough to re-train an error detection system for this ASR.

The paper is organized along the following lines: section 2 presents the related work on error detection task. Section 3 discusses the used features, and particularly proposes to combine different types of word embeddings. Section 4 describes the different neural network architectures proposed in this paper. The experimental setup is described in section 5.

2. RELATED WORK

For two decades, many studies have focused on the ASR error detection task. Recently, several approaches are based on the use of Conditional Random Field (CRF). In [1], authors have focused on detecting error regions generated by Out Of Vocabulary (OOV) words. They proposed an approach based on Conditional 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 ASR errors was presented in [2], which proposes an error detection system based on CRF tagger using various ASR, lexical and syntactic features.

Authors in [3] investigate the application of CRF models as a technique for combining features from different stages of a recognition pipeline. This approach which is founded on using lattice based features of hypotheses, makes it possible to annotate hypotheses with lower word error rate than the 1-best

This work was partially funded by the European Commission through the EUMSSI project, under the contract number 611057, in the framework of the FP7-ICT-2013-10 call.

This work was also partially funded by the French National Research Agency (ANR) through the VERA project, under the contract number ANR-12-BS02-006-01.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

hypothesis of the lattice from which features are extracted.

Recently, a neural network, trained to locate errors in an utterance using a variety of features, was also presented in [4]. This approach needs a complementarity ASR system to extract some features, for effective error detection.

In this paper, we propose to use the CRF approach as a baseline state-of-the-art system, in order to evaluate the performances of our propositions based on neural network architectures and the use of an effective combination of word embeddings built on a huge text corpus.

3. SET OF FEATURES

An error detection system has to attribute the labels *correct* (c) or *error* (e) to each word. This attribution is made by analyzing each recognized word within its context. A set of relevant features must be selected to capture the good information to get a precise classification.

3.1. ASR, lexical and syntactic features

In this study, we nearly use the same features as the one presented in [2], which are detailed as follows:

- ASR features: posterior probabilities generated from the ASR system at the word-level.
- Lexical features: length of the current word (number of letters), 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: POS tag, dependency label (is a grammatical relation holds between a governor (head) and a dependent) and word governor, which are extracted from the transcriptions by using the MACAON NLP Tool chain[1].
- Word: orthographic representation in CRF approaches, as used in [2]. With our neural approach, we will use continuous word representations, which permit us to take advantage of some generalizations extracted during the construction of these word embeddings.

3.2. Word embeddings

Word embeddings are vector representations of words that have been successfully used in several natural language processing tasks [5]. Different approaches have been introduced to calculate word embeddings through neural networks. In this paper, we test three kinds of word embeddings coming from different available implementations. Our goal was to build complementary word embeddings for the ASR task detection. For this task, we need to capture syntactic information in order to use them to analyze sequences of recognized words, but we also need to capture semantic information to measure the relevance of co-occurrences of word in the same ASR hypothesis. We uses the following word embeddings:

- Collobert and Wetson word embeddings revisited by Turian in [6]: they are based on the existence of n-grams, or not, in the training data. In our experiments, we used 5-grams for these word embeddings. Based on our experience, we have got the intuition that these embeddings capture specifically morpho-syntactic similarities. For instance, by visualizing French word vectors in 2D representation, we can notice that words with the same nature (nouns, adjectives, ...), the same gender (male *vs.* female), and the same number (singular *vs.* plural) are projected in the same cluster. Probably because the word position in the n-gram is taken into consideration. These word embeddings are called ***tur*** further in the paper.
- word2vec [7]: we used the model based on the analysis of continuous bag-of-words (CBOW). Like for the *tur* word embeddings, a window size of 5 was chosen. We expect to reach good performances in syntactic modeling by using the CBOW approach as shown in [7], with a more flexible approach than in *tur*. These word embeddings are called ***w2v***.
- GloVe [8]: it is based on the analysis of co-occurrences of words in a window. We chose a window size of 15, in order to capture more semantic information than in the *tur* and *w2v* embeddings. We call these word embeddings ***glove***.

For this study, 100-dimensional word embeddings were computed from a large textual corpus, composed of about 2 billions of words. This corpus was built from articles of the French newspaper "Le Monde", from the French Gigaword corpus, from articles provided by Google News, and from manual transcriptions of about 400 hours of French broadcast news.

In order to take advantage of their complementarity, we propose to combine these three word embeddings. We investigate the use of a denoising auto-encoder [9], denoted *GTW-D*. This auto-encoder is composed of one hidden layer with 100 (GTW-D100) or 200 (GTW-D200) hidden units. It takes as input a corrupted concatenation of the three different embedding vectors and outputs a vector of 300 nodes, and it is trained in order to get in output the denoised concatenation of these three vectors. For each word, the vector of numerical values produced by the hidden layer will be used as the combined *GTW-D* word embedding.

4. NEURAL ARCHITECTURES

We propose to investigate neural networks, well known as one of the most popular classification techniques, which is very effective in a lot of tasks for natural language processing [5].

4.1. Classical MLP

Our first neural system uses a classical MLP composed with one (MLP-1) or two (MLP-2) hidden layers. The number of hidden nodes varies between 100 and 500, according to the error detection performance on the development set. It is fed by the concatenation of the five feature vectors corresponding to the current word, the left and right two words: a window

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

of 5 words is analyzed. The output layer has two nodes corresponding to the labels *c* and *e*.

Neural networks accepting only digital data vectors, features must be represented as numerical values. We identify some non-numeric features (POS tags, dependency labels and word governors), we need to convert them to a digital representation. We propose to use a one-hot representation to replace the POS tags and the dependency labels. For instance, as we use **25** POS tags, we represent the i^{th} POS tag by a 25-dimensional vector, with all its elements equal to 0, except for the i^{th} one, which is equal to 1.

The word governors and the current words are represented by their word embeddings. Figure 1 presents an example of a 252-dimensional feature vector for one word. An input is the concatenation of 5 word feature vectors.

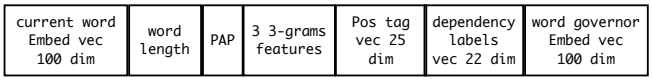

Fig. 1. Neural network input features vector format.

4.2. MLP multi stream

We propose to extend the simple MLP classifier by using multi stream strategy for training the network and injecting the current word feature vector in the last hidden layer.

An MLP multi stream (MLP-MS) architecture is used in order to better integrate the contextual information from neighboring words. This architecture is inspired by [10] where they integrate word and semantic features for theme identification in telephone conversations. The training of the MLP-MS is based on pre-training the hidden layers separately and then fine tuning the whole network. The proposed architecture, very close to the one depicted in Figure 2, is detailed as follows: three feature vectors are used as input to the network. These vectors are respectively the feature vector representing the two left words (L), a feature vector representing the current word (W) and a feature vector for the two right words (R). Each feature vector is used separately in order to train a multilayer perceptron (MLP) with a single hidden layer. The resulting vectors H_{1-L}, H_{1-W} and H_{1-R} are computed with the following equations:

$$H_{1-L} = f(P_{1-L} \times L + b_{1-L}) \quad (1)$$

$$H_{1-W} = f(P_{1-W} \times W + b_{1-W}) \quad (2)$$

$$H_{1-R} = f(P_{1-R} \times R + b_{1-R}) \quad (3)$$

Where P_{1-i} is the weight matrix and b_{1-i} is the bias vector. The first *MLP-MS* hidden layer H_1 corresponds to the concatenation of these three vectors.

The second hidden layer vector H_2 and the output vector O are obtained according to the respective equations :

$$H_2 = g(P_2 \times H_1 + b_2) \quad (4)$$

$$O = q(P \times H_2 + b) \quad (5)$$

where f and g are respectively *relu* and *tanh* activation functions, and q is the *softmax* function.

Furthermore, in order to reinforce the weight of the current word features in the decision process, we propose a variant of the *MLP-MS* architecture, in which the output layer takes as input the concatenation of the second hidden layer vector H_2 and the current word feature vector as follows :

$$O_{inject} = q(P \times (H_2 W) + b) \quad (6)$$

We denote this system *MLP-MS-i* depicted in figure 2

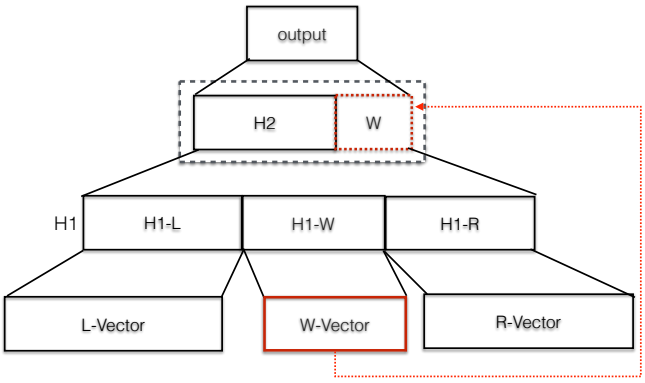

Fig. 2. MLP-MS-i architecture for ASR error detection task.

5. EXPERIMENTS

Results presented as follows were obtained by exploiting three different approaches: a naive one, the CRF one and our propositions. The performance of these approaches is evaluated by using recall (R), precision (P) and F-measure (F) for the erroneous word prediction, and by using global Classification Error Rate (CER) defined as the ratio of the number of misclassifications over the number of recognized words.

5.1. Experimental data

Experimental data are based on the entire official ETAPE corpus [11], composed by audio recordings of French Broadcast News shows with manual transcriptions. This corpus was enriched by automatic transcriptions generated by two different ASR systems:

1. The first LIUM ASR system is a multi-pass system based on the CMU Sphinx decoder, using GMM/HMM acoustic models. This ASR system won the ETAPE evaluation campaign in 2012. A detailed description is presented in [12]. This system will be called ASR1.
2. The second LIUM ASR system is also a multi-pass system and first steps are mainly based on the Kaldi decoder, while last passes (word-lattice rescoring and consensus on network confusion) are similar as the ones used by the first LIUM ASR system. Acoustic models are based on a

DNN/HMM approach. This second ASR system won recently the ASR task of the REPERE evaluation campaign in 2014, about French broadcast news transcription [13]. It is described in [14], and will be called ASR2.

For our experiments, acoustic and language models used on ASR1 and ASR2 have been trained on the same data. Vocabularies are not the same but are close, while linear interpolation coefficients used to build the final composite language models from simple language models are different: they were optimized on different development corpora. Stronger differences between the two ASR systems come from the nature of acoustic models (GMM *vs.* DNN), the search algorithm, the use of full triphones in Kaldi while CMU Sphinx uses inter-word approximations, and from the use of a bigram language model in the first steps of the ASR2 while ASR1 uses a trigram language model.

The automatic transcriptions have been aligned with reference transcriptions using the sclite[2] tool. From this alignment, each word in the corpora has been labeled as correct or incorrect (error). Size, WER and the average error segment size (average span) of the corpora are described in table 1.

System	Name	#words REF	#words HYP	WER
ASR1	Train	349K	316K	25.9
	Dev1	54K	50K	25.2
	Test1	58K	53K	22.5
ASR2	Dev2	54K	50K	23.1
	Test2	58K	53K	20.4

Table 1. Composition of the experimental corpus.

5.2. Comparison of different word embeddings

To evaluate the impact of the different types of word embeddings on the ASR error detection task, we test them with the *MLP-2* system. In table 2, we observe that our proposition to combine word embeddings is helpful and yields significant improvement, especially when using the denoising auto-encoder with 200 hidden units.

		Label error			Global
NN	**Embed**	**P**	**R**	**F**	**CER**
MLP-2	glove	68.7	53.1	59.9	10.56
	w2v	69.2	54.7	61.1	10.36
	tur	70.3	53.0	60.4	10.32
	GTW-D100	69.9	55.3	61.8	10.18
	GTW-D200	70.0	56.4	62.5	**10.07**

Table 2. Comparison on Dev1 of different types of word embeddings used as features in MLP-2 error detection system.

[2]http://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm

The *GTW-D200* word embeddings are used for the remaining experiments.

5.3. Comparison of different neural architectures

A set of experiments was performed using the different neural network architectures, compared with the two following systems:

- Naive: the naive approach consists in applying a threshold on the word posterior probabilities to predict whether the word is correct or incorrect. The threshold that gives a lower CER on the Dev1 is applied to compute the CER on Test1 and Test2. The best threshold found is 0.79.
- CRF: this is a state-of-the-art system based on the CRF tagger provided by Wapiti[3] applied on the set of features presented in section 3.

All the systems were trained on the train corpus from ASR1 outputs, and were applied on the ASR1 outputs on the development and test corpora, then on the ASR2 outputs on the same corpora without re-training, nor adaptation. Results reached by all the systems are summarized in table 3 when applied on the ASR1 outputs and table 4 on the ASR2 hypotheses.

		Label error			Global
corpus	**approach**	**P**	**R**	**F**	**CER**
Dev1	Naive	66.9	48.8	56.4	11.21
	CRF	70.8	50.6	59.0	10.44
	MLP-1	71.0	53.3	60.9	10.17
	MLP-2	70.0	56.4	62.5	10.07
	MLP-MS	70.7	55.9	62.5	**9.99**
	MLP-MS-i	68.8	58.0	**63.0**	10.15
Test1	Naive	65.3	47.1	54.7	9.42
	CRF	69.2	49.3	57.6	8.78
	MLP-1	69.3	53.3	60.3	8.50
	MLP-2	67.8	56.3	61.5	8.52
	MLP-MS	68.8	55.5	61.4	**8.43**
	MLP-MS-i	67.5	57.4	**62.1**	8.49

Table 3. Error detection results on ASR1 transcriptions.

As expected, the state-of-the-art CRF approach obtains better results than the naive approach. These results are improved with the use of neural architectures. Considering the MLP-2 system, a CER reduction of 3% is observed on Test1 and Test 2 in comparison with the CRF approach. MLP-MS improves these performances, with a reduction of 4% in comparison with the CRF baseline system on the Test1 corpus, and 3.5% on Test2.

The injection of the current word feature vector in the output layer (MLP-MS-i) is helpful on Dev2 and Test2, which yields respectively 5.8% and 5.1% of CER reduction in comparison to the CRF approach: this approach is the most robust

[3]http://wapiti.limsi.fr

		Label error			Global
corpus	**approach**	**P**	**R**	**F**	**CER**
Dev2	Naive	68.6	31.0	42.7	10.95
	CRF	63.6	40.5	49.5	10.88
	MLP-1	70.3	35.9	47.6	10.43
	MLP-2	68.0	38.4	49.1	10.49
	MLP-MS	69.8	36.5	47.9	10.44
	MLP-MS-i	68.3	41.3	**51.5**	**10.25**
Test2	Naive	69.3	32.2	43.9	8.70
	CRF	64.3	42.6	51.3	8.59
	MLP-1	69.4	37.0	48.3	8.41
	MLP-2	68.2	40.0	50.4	8.34
	MLP-MS	70.0	38.2	49.4	8.29
	MLP-MS-i	68.5	42.9	**52.7**	**8.15**

Table 4. Error detection results on ASR2 transcriptions.

to the ASR system. More, in terms of F-measure applied on the misrecognized words, the MLP-MS-i approach reaches always the best results, whatever the ASR system.

6. CONCLUSION

In order to improve error detection in ASR outputs, we have proposed a new approach based on neural network architectures that combine a set of features (ASR-based, lexical and syntactic) and word embeddings to represent a word and its context. We have also proposed an effective approach to combine different word embeddings by using a denoising autoencoder. Experiments performed on the automatic transcriptions of ETAPE corpus generated from two different ASR systems led to significant improvements, in comparison with a state-of-the art CRF approach. The MLP-MS-i approach is the most robust to the ASR system, reaching the best results in terms of CER when applied on the outputs of an ASR system different to the one used to train the error detection system. Last, in terms of F-measure applied on the misrecognized words, the MLP-MS-i approach reaches always the best results, whatever the ASR system.

REFERENCES

[1] C. Parada, M. Dredze, D. Filimonov, and F. Jelinek, "Contextual information improves oov detection in speech," *in North American chapter of thes Association for Computational Linguistics (NAACL)*, 2010.

[2] Frédric Béchet and Benoit Favre, "ASR error segment localisation for spoken recovery strategy," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference*, 2013.

[3] M. S. Seigel and P. C. Woodland, "Combining information sources for confidence estimation with crf models," in *INTERSPEECH*, 2011, pp. 905–908.

[4] Tam Yik-Cheung, Yun Lei, Jing Zheng, and Wen Wang, "ASR error detection using recurrent neural network language model and complementary ASR," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 2312–2316.

[5] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, "Natural Language Processing (Almost) from Scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, 2011.

[6] Joseph Turian, Lev Ratinov, and Yoshua Bengio, "Word representations: A simple and general method for semisupervised learning," *In ACL*, pp. 384–394, 2010.

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," in *Proceedings of Workshop at ICLR*, 2013.

[8] Jeffrey Pennington, Richard Socher, and Christopher D Manning, "Glove: Global vectors for word representation," in *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 2014, vol. 12.

[9] P. Vincent, H. Larochelle, Y. Bengio, and P.A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008.

[10] Yannick Estève, Mohamed Bouallegue, Carole Lailler, Mohamed Morchid, Richard Dufour, Georges Linarès, Driss Matrouf, and Renato De Mori, "Integration of word and semantic features for theme identification in telephone conversations," in *6th International Workshop on Spoken Dialog Systems (IWSDS 2015)*, 2015.

[11] 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.

[12] 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.

[13] Aude Giraudel, Matthieu Carré, Valérie Mapelli, Juliette Kahn, Olivier Galibert, and Ludovic Quintard, "The REPERE Corpus: a multimodal corpus for person recognition," in *LREC*, 2012, pp. 1102–1107.

[14] Anthony Rousseau, Gilles Boulianne, Paul Deléglise, Yannick Estève, Vishwa Gupta, and Sylvain Meignier, "LIUM and CRIM ASR System Combination for the REPERE Evaluation Campaign," in *Text, Speech and Dialogue*. Springer, 2014, pp. 441–448.