Which ASR errors are hard to detect?

Sahar Ghannay, Nathalie Camelin, Yannick Estève

LIUM-University of Le Mans, France

firstname.lastname@univ-lemans.fr

Abstract

In this paper, we focus on error detection in Automatic Speech Recognition (ASR) outputs. We present a new approach using continuous word representation (word embeddings) through a neural network classifier. This classifier is in charge to attribute a label (*error* or *correct*) for each word within an ASR hypothesis. Combining with word embeddings, inputs are based on a set of features (ASR confidence scores, lexical, and syntactic features, including contextual information from each word).

Experiments were conducted on the automatic transcriptions generated by the LIUM ASR system applied on the ETAPE corpus (French broadcast news). They show that the proposed neural architecture outperforms the state-of-the-art approach based on the use of Conditional Random Fields (CRF).

Particularly in this study, we are interested in the analysis of the classifier outputs, in order to perceive the errors that are hard to detect. Results of this analysis are presented in this paper, providing useful information in order to improve the proposed ASR error detection system.

Index Terms: ASR error detection, neural networks, continuous word representations, analysis of the ASR error detection system outputs.

1. Introduction

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

Error detection aims to improve the exploitation of ASR outputs by downstream applications, but it is a difficult task because there are several types of errors, which can range from a simple mistake on the number agreement 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 order to circumvent these problems it is vital to employ reliable confidence measures and combine them with a classifier in a reasonable way to identify ASR errors.

In this study, we propose a neural architecture to detect ASR errors. First, we focus on the combination of ASR confidence scores, lexical, syntactic and a continuous word representation, including contextual information from each word within a neural network classifier for effective misrecognized word detection. Second, we perform the analysis of the classifier outputs. Thus, we have the dual purpose of studying the performance of our system (both features and architecture sides) and knowing which ASR errors are the hardest to detect.

In section 2, we first review prior works related to ASR error detection and then recent studies about continuous word representation, which is used by our ASR error detection system. In section 3, we describe the proposed set of features, while the error detection system is presented in section 4. The experimental setup, results, and analyses are presented in section 5, just before the conclusion.

2. Related work

For two decades, many studies have focused on the ASR error detection task. Recently, several approaches were based on the use of Conditional Random Field (CRF). Authors in [1] have focused on detecting error regions generated by Out Of Vocabulary (OOV) words. They proposed 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 was presented in [2], which proposes an error detection system based on a CRF tagger using ASR, lexical and syntactic features. In [3], new features gathered from other knowledge sources than the decoder itself were explored for ASR error detection, which are a binary feature that compares the outputs from two different ASR systems (word by word), a feature based on the number of hits of the hypothesized bigrams, obtained by queries entered into a very popular Web search engine. The integration of these features within a maximum entropy and CRF models, led to significant improvements compared to a baseline using only decoder-based features.

Recently, a neural network trained to locate errors in an utterance using a variety of features was also presented in [4]. Some of these features are captured from forward and backward recurrent neural network language model in order to capture long distance word context within and across previous utterances. The other features are extracted from two complementary ASR systems.

Continuous word representations are successfully used in several Natural Language Processing (NLP) tasks as additional features. Authors in [5] have evaluated different types of word representations namely, Brown clusterings, the Collobert and Wetson embeddings [6] and the hierarchical log-bilinear (HLBL) [7] embeddings of words, and their combination by a simple concatenation, on chunking and named entity recognition task.

In this paper, we propose to integrate ASR confidence scores, lexical, syntactic and the continuous word representation, including contextual information from each word in a neural network architecture designed for ASR error detection.

3. Set of features

In this section, we describe the features we collected for each word and how the features were extracted. Some of these features are nearly the same as the ones presented in [2]. The word feature vector is the concatenation of the features detailed in the next sub-sections:

3.1. ASR confidence scores

ASR confidence scores are the posterior probabilities generated from the ASR system (PAP). The word posterior probability is computed over confusion network, which is approximated by the sum of the posterior probabilities of all transitions through this word and that compete with it.

3.2. Lexical features

Lexical features are derived from the word hypothesis output by the ASR system. They include: the word length that represents the number of letters in the 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.

3.3. Syntactic features

We obtain syntactic features by automatically assigning part-of-speech tags (POS tags), dependency labels and word governors using the MACAON NLP Tool chain[1] to process the ASR outputs.

POS tags, dependency labels and words must be converted to a digital representation in order to be used as inputs of a neural network. We propose to use a one-hot representation to replace the POS tags and the dependency labels. For instance, since 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.

3.4. Word

The orthographic representation is used in CRF system. In the neural one, we will use a 200-dimension vector named *GTW-D200*. This word embedding is the result of the combination, using a denoising auto-encoder, of three different continuous word embeddings: a variant of the Collobert and Wetson word embeddings [5], word2vec [8] and GloVe [9]. In that way, we take advantage of: firstly, the power of generalization of embeddings and, secondly the complementarity of different kind of embeddings.

Three 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.

The denoising auto-encoder is composed of one hidden layer with 200 hidden units. It takes as input the concatenation of the three different embedding vectors and outputs a vector of 300 nodes. For each word, the vector of numerical values produced by the hidden layer will be used as the combined word embeddings.

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

4. Error prediction system

The error prediction system has to attribute the label *correct* or *error* to each word based on the set of features described in section 3. This attribution is made by analyzing each recognized word within its context. The context window size used in this study is 2 on either side of the current word.

The proposed system is based on a multi-stream strategy to train the network, named multilayer perceptron multi stream (MLP-MS). The MLP-MS architecture is used in order to better integrate the contextual information from neighboring words. This architecture is inspired by [11] 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, depicted in Figure 1, 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), the feature vector representing the current word (W) and the 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. Formally, the architecture is described by the following equations:

$$H_{1,X} = f(P_{1,X} \times X + b_{1,X}) \quad (1)$$

where X represents respectively the three feature vectors (L,W and R), $P_{1,X}$ is the weight matrix and $b_{1,X}$ is the bias vector. The resulting vectors $H_{1,L}$, $H_{1,W}$ and $H_{1,R}$ are concatenated and this concatenated resulting vector H_1 is presented as the input of the second *MLP-MS* hidden layer H_2 computed according to the equation:

$$H_2 = g(P_2 \times H_1 + b_2) \quad (2)$$

Last, the output layer is a vector O_k of k=2 nodes corresponding to the 2 labels $correct$ and $error$:

$$O_k = q(P_O \times H_2 + b_O) \quad (3)$$

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

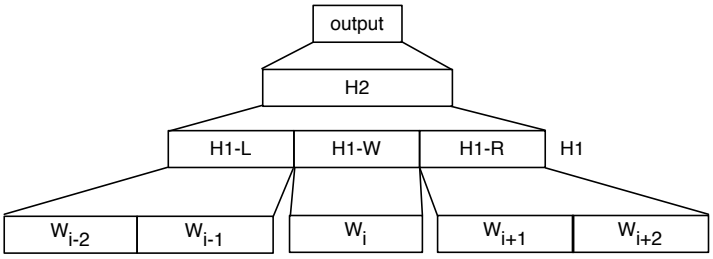

Figure 1: MLP-MS architecture for ASR error detection task.

The Theano library [12] was used to train MLP-MS system reported in this paper.

5. Experiments

5.1. Experimental data

Experimental data are based on the entire official ETAPE corpus [13], composed by audio recordings of French Broadcast News shows with manual transcriptions. This corpus was enriched

by automatic transcriptions generated by the LIUM ASR system, which 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 [14].

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 (C) or error (E), called in the remainder of the paper ground truth. The description of the experimental data is reported in Table 1.

Name	#words ref	#words hyp	WER
Train	349K	316K	25.9
Dev	54K	50K	25.2
Test	58K	53K	22.5

Table 1: Description of the experimental corpus.

5.2. Experimental results

This section reports the experimental results made on the data set using the ASR error detection system *MLP-MS*. The performance of the proposed approach is compared with a state-of-the-art system based on the CRF tagger provided by Wapiti[3] applied to the set of features presented in section 3.

The performance 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.

5.2.1. Comparison of different word representations

A set of experiments is performed in order to evaluate the impact of the different types of word embeddings as well as their combination. The ASR error detection system is trained on the train corpus and is applied on the development set Dev. Experimental results, presented in table 2, show that our proposition to combine word embeddings is helpful and yields significant improvement in terms of *CER* compared to the use of the single word embeddings.

		Label error			Global
Approach	**Representations**	**P**	**R**	**F**	**CER**
Neural	glove	69.04	51.25	58.83	10.66
	w2v	67.00	**57.37**	61.81	10.54
	tur	69.67	51.34	59.11	10.56
	GTW-D200	**70.74**	55.93	**62.47**	**9.99**
CRF	discrete	70.8	50.6	59.0	10.44

Table 2: Comparison on Dev of different types of word embeddings used as features in the ASR error detection system.

In the remainder of this paper, *MLP-MS* always refers to the architecture presented in section 4 used with the *GTW-D200* word embeddings, and with right and left context of 2 words each. Table 3 shows the performance of the *MLP-MS* system applied to the Test corpus. This representation yields 4% of CER reduction comparing to the state-of-the-art CRF approach.

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

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

	Label error			Global
Approach	**P**	**R**	**F**	**CER**
CRF	69.2	49.3	57.6	8.78
MLP-MS	68.8	**55.5**	**61.4**	**8.43**

Table 3: Error detection results on Test corpus

5.2.2. Comparison of context size

Table 4 reports the experimental results made on different context sizes (0 and 1). For *MLP W/O context*, a multilayer perceptron with one hidden layer is used. It takes as input only the feature vector of the current word. For *MLP-MS 1w context*, the MLP-MS sytem is used. It takes as input a window of size 3 including 3 feature vectors for the current, left and right words.

As proven in table 4, the choice of context size of 2 words on either side is helpful, comparing to the use of context of size 0 or 1. The precision for the erroneous word prediction increases with the size of the context respectively 67.32%, 68.87% and 70.74% for a right and left context of 0, 1 and 2 words. The same improvement is observed for CER. These results show that information about the neighborhood of a recognized word is really useful to detect an error.

	Label error			Global
Architecture	**P**	**R**	**F**	**CER**
MLP W/O context	67.32	55.88	61.07	10.59
MLP-MS 1w context	68.87	53.32	60.10	10.52
MLP-MS	70.74	55.93	62.47	9.99

Table 4: Comparison on Dev of different context size : right and left context of 0 and 1 word.

5.3. Analysis of the ASR error detection system outputs

We report in this section the analysis made for the ASR error detection system outputs called *predictions* in the following. We seek to study the performance of the *predictions* compared to the *ground truth*, based on several categories.

5.3.1. Word length analysis

As shown in figure 2, the words of length 2 and 3 have the lowest precision and recall, given that, this words represent respectively 28% and 18% of the erroneous words on Dev. We observe also the same performance for the words of length 12 and 14, but this words are not frequent and they represent almost 0.44% of the erroneous words on Dev.

5.3.2. Function and non function words analysis

We define a list of 160 function words (the, is , at,) which are the most frequent in the corpus. We can notice in table 5 that the ASR error detection system performs better on non function words than on function ones. Furthermore, we noticed that 75.65% of erroneous function words have a length of 2 or 3 characters. Thus the results observed totally match those of previous sub-section 5.3.1.

Even if non function word are better detected than function words, 61% of recall for error detection is not satisfying for such important words, which carry the meaning of the discourse. As it is difficult to capture semantic anomalies between words in a small context window, such anomalies should be easier to detect

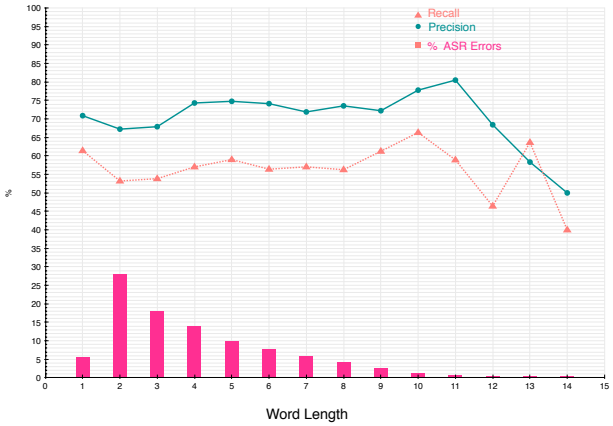

Figure 2: Recall and precision for the erroneous word prediction and the percentage of erroneous words by word length on Dev corpus.

Words	Label error	
	P	R
Non function	75.1	61.0
Function	66.9	51.7

Table 5: Function and non function words analysis on Dev corpus

by analyzing non function words in a more global context. For instance, by integrating additional semantic information about the document, the video, the show...

5.3.3. *Average error segment size (average span) analysis*

Table 6 presents the average span and the standard deviation for the *ground truth*, the *predictions*, the *correct predictions* and the *CRF* outputs. We define the average span of the *correct predictions* as the average error segment of the contiguous errors correctly detected. We observe that the average span of CRF outputs is nearly the same as *ground truth*. However, for both the *predictions* and the *correct predictions* the average span is smaller by respectively 9.88% and 17.6% compared to the *ground truth*. In addition, the standard deviation of the *predictions* is 23.75% larger than the *ground truth*. This gap related to the error segment size between the *ground truth*, the *predictions* and the *correct predictions* is due to the architecture of the proposed ASR error detection system. This one takes only local decisions and is not currently designed to perform optimally sequence prediction while CRF seems to be able to better capture such information. In the future, we will adapt our architecture to model *a priori* information about the error segment size, for an efficient use of the contextual information. This may be experimented by using a recurrent neural network architecture for instance, or by using a two-pass strategy.

5.3.4. *Current word context analysis*

Considering that the MLP-MS system takes as input a window of size 5, including the current word and its context of size 2 on either side, we look to study the *predictions* behavior compared to the *ground truth*, depending on the number of errors in the context (from 0 to 4 errors). Figure 3 illustrates the precision, recall and F-measures for the erroneous word prediction relative

	Corpus	Average span	Standard deviation
Ground truth	Train	3.03	1.72
	Dev	3.24	2.15
Predictions	Dev	2.92	2.82
Correct Predictions	Dev	2.67	1.17
CRF	Dev	3.29	1.81

Table 6: The average span and the standard deviation for the *ground truth*, the *predictions*, the *correct predictions* and the *CRF* outputs.

to the number of errors in its context.

We observe that, when there are 1, 2 or 3 errors in the context, the classifier achieves respectively 64.3%, 72.6%, and 77.3% of precision for *correct predictions* and 50%, 53.7%, and 51.4% of recall. When the context is completely erroneous the system is more accurate by 93.2% of precision. However the classifier has difficulty to detect isolated errors *i.e.* 0 errors in the context. This maybe can be explained by the fact that the major part of isolated erroneous recognized words do not trigger a significant linguistic rupture which could be detected by the error detection system. Indeed, ASR systems use languages models in order to propose the more acceptable (and probable) sequences of word. When an ASR system makes an isolated error, this is due to the fact that this error does not disrupt the language model. If the language model is disrupted, this implies a chain of errors, easier to detect.

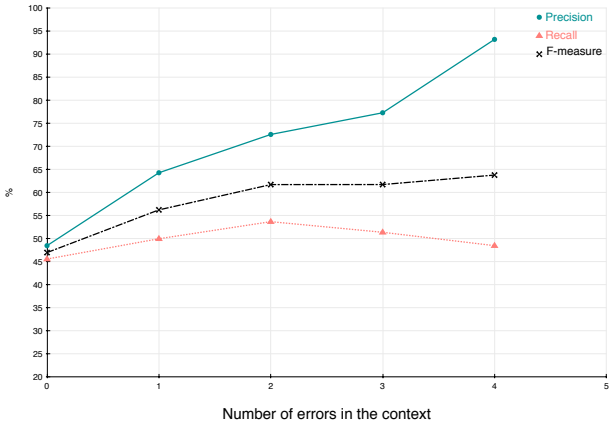

Figure 3: Precision, recall and F-measures for the erroneous word prediction *error* relative to the number of errors in its context.

5.3.5. *Syntactic role analysis*

This analysis is based on the syntactic role (POS tag) of the word within the whole sentence (automatic transcriptions and reference transcriptions). Here, we are interested on the behavior of the system when the hypothesis word and the reference word have the same (EQ) or different (DIFF) POS tags.

Table 7 shows the results according to these two cases (equal or different POS tags). We observe that the system is more accurate when the POS tags are different, it achieves 95.57% of precision for the erroneous word prediction. When

the POS tags are equal, the system clearly does not perform well.

These results reinforce the idea that weak linguistic disruption makes ASR errors hard to detect.

	Label error		Global
POS	**P**	**R**	**CER**
EQ	29.01	51.51	5.05
DIFF	95.57	56.82	38.69

Table 7: Error analysis results on Dev corpus according to the part of speech tag of the automatic transcriptions and reference transcriptions

Figure 4 illustrates the number of current words having the label *error* and equal POS tag relative to the number of errors in its context having different POS tags, for the *ground truth*, the *predictions* and the *correct predictions*. As shown in these figures, the low precision is related to the high number of predicted errors (*predictions*), which is almost twice compared to the *ground truth* when there are up to 3 errors in context. This means that the syntactic information is not sufficient to distinguish recognized and misrecognized words. Moreover, the pos tagging of erroneous words within a sentence can biased the syntactic information, and can prevent the system to take reliable decisions.

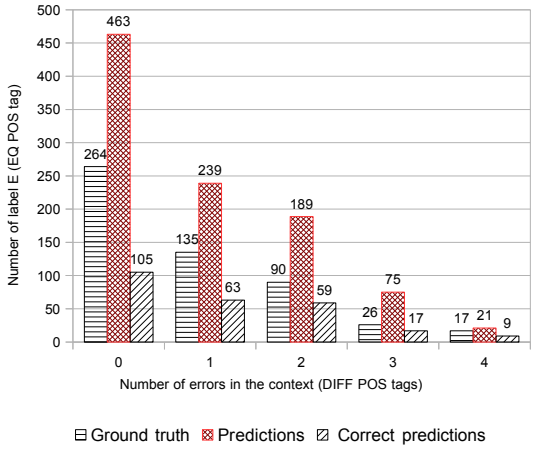

Figure 4: The number of current words having the label *error* and equal POS tag relative to the number of errors in its context with different POS tags

6. Conclusion

Considering the framework of ASR error detection, we present a new approach using continuous word representation (word embeddings) through a neural network classifier. This classifier combines, with word embeddings, a set of features (ASR confidence scores, lexical and syntactic) to represent a word and its context.

Experiments performed on the automatic transcriptions generated by the LIUM ASR system applied to the ETAPE corpus (French broadcast news) led to significant improvements, in comparison with a state-of-the art CRF approach.

Based on the analyses made on the classifier outputs, some avenues to explore are emphasized in order to improve the ASR error detection. First, it seems important to take into account *a priori* information about error behavior in terms of propagation. Even if the proposed approach provides globally better results than CRF, the latter are more relevant to detect erroneous islands. To handle this phenomenom, a recurrent neural network (RNN) will be explore in the future.

Then, isolated errors are difficult to detect, like errors which do not change the POS of the erroneous recognized word: weak linguistic disruption makes ASR errors hard to detect. New strategies and new information have to be integrated in future work. For instance, by adding global semantic information about the document, the video, the show in order to make easier the detection of a semantic anomaly.

Therefore, it is interesting to add acoustic and prosodic features as complementary of syntactic features for effective error detection. This can be motivated by previous researches [15], [16] and [17]. Stoyanchev et al.

All these propositions (RNN, semantics, acoustics, prosody) will be explored in future work.

7. Acknowledgements

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, by the French National Research Agency (ANR) through the VERA project, under the contract number ANR-12-BS02-006-01, and by the Région Pays de la Loire.

8. References

[1] C. Parada, M. Dredze, D. Filimonov, and F. Jelinek, "Contextual information improves oov detection in speech," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010.

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

[3] T. Pellegrini and I. Trancoso, "Improving asr error detection with non-decoder based features." in *INTERSPEECH*, 2010, pp. 1950–1953.

[4] T. Yik-Cheung, Y. Lei, J. Zheng, and W. 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] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: A simple and general method for semisupervised learning," 2010, pp. 384–394.

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," vol. 12. JMLR.org, 2011, pp. 2493–2537. [Online]. Available: http://dl.acm.org/citation.cfm?id=1953048.2078186

[7] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 1081–1088. [Online]. Available: http://papers.nips.cc/paper/3583-a-scalable-hierarchical-distributed-language-model.pdf

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of Workshop at ICLR*, 2013.

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

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

[11] Y. Estève, M. Bouallegue, C. Lailler, M. Morchid, R. Dufour, G. Linarès, D. Matrouf, and R. D. Mori, "Integration of word and semantic features for theme identification in telephone conversations," in *6th International Workshop on Spoken Dialog Systems (IWSDS 2015)*, 2015.

[12] 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.

[13] G. Gravier, G. Adda, N. Paulsson, M. Carr, A. Giraudel, and O. 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.

[14] P. Deléglise, Y. Estève, S. Meignier, and T. 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.

[15] S. Stoyanchev, P. Salletmayr, J. Yang, and J. Hirschberg, "Localized detection of speech recognition errors," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*, Dec 2012, pp. 25–30.

[16] J. Hirschberg, D. Litman, and M. Swerts, "Prosodic and other cues to speech recognition failures," *Speech Communication*, vol. 43, no. 1, pp. 155–175, 2004.

[17] S. Goldwater, D. Jurafsky, and C. D. Manning, "Which words are hard to recognize? prosodic, lexical, and disfluency factors that increase speech recognition error rates," *Speech Communication*, pp. 181–200, 2010.