Utilisation des représentations continues des mots et des paramètres prosodiques pour la détection d'erreurs dans les transcriptions automatiques de la parole

Sahar Ghannay [1] Yannick Estève [1] Nathalie Camelin [1] Camille Dutrey [2, 3]
Fabián Santiago [2] Martine Adda-Decker [2, 4]
(1) Laboratoire d'Informatique de l'Université du Maine (LIUM), Avenue Laennec, Le Mans, France
(2) Laboratoire de Phonétique et Phonologie (LPP), 19 rue des Bernardins, Paris, France
(3) Laboratoire National de Métrologie et d'Essais (LNE), 29 avenue Roger Hennequin, Trappes, France
(4) Laboratoire d'informatique pour la mécanique et les sciences de l'ingénieur (LIMSI), Rue John Von Neumann, Orsay, France
{prenom.nom}@univ-lemans.fr[1] , {prenom.nom}@univ-paris3.fr[2]

RÉSUMÉ

Récemment, l'utilisation des représentations continues de mots a connu beaucoup de succès dans plusieurs tâches de traitement du langage naturel. Dans cet article, nous proposons d'étudier leur utilisation dans une architecture neuronale pour la tâche de détection des erreurs au sein de transcriptions automatiques de la parole. Nous avons également expérimenté et évalué l'utilisation de paramètres prosodiques en suppléments des paramètres classiques (lexicaux, syntaxiques, ...). La principale contribution de cet article porte sur la combinaison de différentes représentations continues de mots : plusieurs approches de combinaison sont proposées et évaluées afin de tirer profit de leurs complémentarités. Les expériences sont effectuées sur des transcriptions automatiques du corpus ETAPE générées par le système de reconnaissance automatique du LIUM. Les résultats obtenus sont meilleurs que ceux d'un système état de l'art basé sur les champs aléatoires conditionnels. Pour terminer, nous montrons que la mesure de confiance produite est particulièrement bien calibrée selon une évaluation en terme d'Entropie Croisée Normalisée (NCE).

ABSTRACT

Combining continuous word representation and prosodic features for ASR error detection

Recent advances in continuous word representation have been successfully used in several natural language processing tasks. This paper focuses on error prediction in Automatic Speech Recognition (ASR) outputs and proposes to investigate the use of continuous word representation (word embeddings) within a neural network architecture. The main contribution of this paper is about word embeddings combination : several combination approaches are proposed in order to take advantage of their complementarity. The use of prosodic features, in addition to classical syntactic ones, is evaluated. Experiments are made on automatic transcriptions generated by the LIUM ASR system applied on the ETAPE corpus. They show that the proposed neural architecture, using an effective continuous word representation combination and prosodic features as additional features, outperforms significantly state-of-the-art approach based on the use of Conditional Random Fields. Last, the proposed system produces a well calibrated confidence measure, evaluated in terms of NCE.

MOTS-CLÉS : Détection des erreurs de SRAP, réseau de neurones, représentation continue de mot, paramètres prosodiques.

KEYWORDS: ASR error detection, neural networks, word embeddings, prosodic features.

1 Introduction

Les avancées scientifiques récentes dans le domaine du traitement de la parole ainsi que la disponibilité d'une puissance de calcul croissante, ont conduit à l'obtention de performances très intéressantes d'un point de vue applicatif dans le domaine de la reconnaissance de la parole (SRAP). Cependant, malgré ces performances, les SRAPs génèrent encore des erreurs. Cela s'explique par leur sensibilité aux diverses variabilités liées à l' environnement acoustique, au locuteur, au style de langage, à la thématique du discours, *etc*). Ces erreurs présentent un obstacle à l'exploitation des transcriptions automatiques, par exemple pour certains traitements automatiques tels que l'extraction d'information, la traduction de la parole, la compréhension de la parole, *etc*.

L'exploitation efficace des transcriptions automatiques reste un but qui peut être atteint si l'on est capable de détecter et/ou de corriger des erreurs contenues dans les transcriptions automatiques. Cependant, la détection d'erreurs n'est pas une tâche facile, du fait qu'il existe plusieurs types d'erreurs. Ces erreurs peuvent aller de la simple substitution d'un mot par un homophone à l'insertion d'un mot non pertinent pour la compréhension globale de la séquence de mots. Elles peuvent aussi se répercuter sur les mots voisins et entraîner une séquence de mots erronés.

Dans cet article, nous proposons une architecture neuronale pour la détection d'erreurs. Nous nous intéressons à l'utilisation des représentations continues des mots [1], qui ont été introduites à l'origine pour la construction de modèles de langages neuronaux (Schwenk *et al.*, 2006). Ces représentations ont montré leurs impacts positifs dans de nombreuses tâches de traitement automatique du langage naturel (NLP) telles que : l'étiquetage morpho-syntaxique, le regroupement en syntagmes, la reconnaissance d'entités nommées ou encore l'étiquetage de rôles sémantiques (Turian *et al.*, 2010; Collobert *et al.*, 2011).

Dans le cadre de la tâche de détection d'erreurs de SRAP, nous étudions l'utilisation de plusieurs types de word embeddings provenant de différentes implémentations disponibles : `word2vec` (Mikolov *et al.*, 2013), `GloVe` (Pennington *et al.*, 2014) et une variante des embeddings de Collobert et Weston (Turian *et al.*, 2010). Afin de bénéficier de leurs potentielles complémentarités, nous proposons différentes approches de combinaison.

Le meilleur embedding obtenu par combinaison est utilisé pour représenter un mot reconnu, ainsi que les probabilités *a posteriori* de ce mot, des paramètres lexicaux, syntaxiques et prosodiques, qui sont intégrés dans une architecture neuronale pour une détection efficace des erreurs. Bien que les paramètres prosodiques aient déjà été utilisés dans le cadre de la détection des erreurs au niveau de l'énoncé (utterance), leur utilisation pour la détection des erreurs au niveau du mot a été moins étudiée.

Enfin, nous nous intéresserons à l'évaluation des mesures de confiance produites par le système neuronal.

Cet article est organisé de la manière suivante : la section 2 présente les travaux liés à la tâche de détection des erreurs, l'intégration des embeddings et des paramètres prosodiques pour cette tâche. La section 3 détaille les différents types d'embeddings ainsi que les approches proposeés pour les combiner. La section 4 décrit le système de détection d'erreurs proposé. Le protocole expérimental et les résultats sont décrits dans la section 5, juste avant la conclusion (Section 6).

1. par la suite on utilisera la terminologie anglaise *word embeddings*

2 Travaux connexes

Depuis deux décennies, de nombreuses études se focalisent sur la détection des erreurs de SRAP.

Plusieurs approches sont fondées sur l'utilisation des champs aléatoires conditionnels (CRF). Dans (Parada *et al.*, 2010), les auteurs se sont intéressés à la détection des régions d'erreurs générées par les mots hors vocabulaires en prenant en compte des informations contextuelles des régions. Une approche similaire pour d'autres types d'erreurs a été présentée dans (Béchet & Favre, 2013). Celle-ci est basée sur un étiqueteur de séquence à base de CRF utilisant des paramètres issus de systèmes de transcription, des paramètres lexicaux et syntaxiques.

L'approche la plus récente utilise un réseau de neurones qui intègre plusieurs sources d'information afin de détecter si un mot est correct ou erroné (Yik-Cheung *et al.*, 2014). On peut notamment citer des paramètres extraits à partir du modèle de langage basé sur les réseaux de neurones récurrents, des réseaux de confusion. D'autres paramètres proviennent également de la complémentarité de deux SRAPs.

Les embeddings constituent une projection des mots du vocabulaire dans un espace de faible dimension. Ils sont utilisés avec succès, comme paramètres supplémentaires, dans plusieurs tâches NLP (Turian *et al.*, 2010; Collobert *et al.*, 2011). Les auteurs dans (Turian *et al.*, 2010) ont évalué différents types de word embeddings ainsi que leur combinaison par simple concaténation pour la tâche de reconnaissance d'entités nommées et le regroupement en syntagmes.

Notre utilisation des paramètres prosodiques est motivée par de précédents travaux, notamment (Stoyanchev *et al.*, 2012) et (Goldwater *et al.*, 2010). Les premiers ont montré que la combinaison des paramètres prosodiques et syntaxiques est utile pour localiser les mots mal reconnus dans un tour de parole. Les seconds ont découvert que les mots mal reconnus ont des valeurs prosodiques extrêmes.

Dans cette étude, nous proposons d'intégrer la meilleure combinaison des embeddings avec d'autres paramètres supplémentaires dans une architecture neuronale conçue pour la détection des erreurs de SRAP.

3 Représentations continues (*embeddings*) de mot

Différentes approches ont été introduites pour calculer les embeddings de mots à travers les réseaux de neurones.

Dans le cadre de la détection des erreurs, nous avons besoin de capturer des informations syntaxiques afin de les utiliser pour analyser les séquences de mots reconnus, mais nous avons aussi besoin de capturer des informations sémantiques pour mesurer la pertinence des co-occurrences de mots dans la même hypothèse. Nous avons utilisé et évalué, trois types d'embeddings provenant de différentes implémentations disponibles (plus de détails dans (Ghannay *et al.*, 2015a)).

3.1 Description des embeddings de mots

Trois types d'embeddings, à 100 dimensions chacun, ont été calculés à partir d'un vaste corpus textuel composé d'environ 2 milliards de mots. Ce corpus a été construit à partir des articles du journal français *Le Monde*, le corpus *Gigaword*, les articles fournis par *Google News* et les transcriptions manuelles d'environ 400 heures d'émissions françaises.

Ces embeddings sont détaillés dans ce qui suit :

tur : embeddings de Collobert et Weston (Turian *et al.*, 2010), revisités par Joseph Turian. Ils

sont basés sur l'existence ou non de n-grammes dans le corpus d'apprentissage.

w2v-CBOW : calculés avec la boite à outils *word2vec*. Ils sont estimés avec l'approche sac de mots continus (*Continuous bag-of-words (CBOW)*).

GloVe : basés sur l'analyse des co-occurrences des mots dans une fenêtre (Pennington *et al.*, 2014).

3.2 Combinaison des embeddings

Afin de tirer profit de la complémentarité des embeddings décrits ci-dessus, nous avons proposé de les combiner en utilisant plusieurs approches détaillées ci-après.

3.2.1 Concaténation simple

La première approche est inspirée de celle proposée par (Turian *et al.*, 2010). Elle consiste à concaténer les embeddings selon cet ordre : *GloVe*, *tur* et *w2v-CBOW*, (nommé ***GTW***). Chaque mot est ainsi représenté par un vecteur de taille 300.

3.2.2 Analyse en Composantes Principales (ACP)

Cette deuxième approche consiste à transformer des variables corrélées entre elles en nouvelles variables décorrélées les unes des autres (appelées composantes principales ou axes principaux). Les premiers axes portent plus d'informations que les derniers (en terme de dispersion de données). L'ACP est appliquée au vecteur *GTW* (celui obtenu par simple concaténation). Elle est calculée en utilisant la matrice de corrélation pour obtenir le nouveau système vectoriel. Ce système est projeté ensuite dans une nouvelle base. Nous considérons par la suite uniquement les 200 premières composantes du vecteur projeté (nommé ***GTW-PCA-200***).

3.2.3 Auto-encodeurs

La troisième approche se base sur l'utilisation d'auto-encodeurs ordinaire (*O*) et de débruitage (*D*). Ces auto-encodeurs sont composés d'une couche cachée contenant 200 (*GTW-200*) unités cachées chacune. Ils prennent en entrée le vecteur *GTW* et génèrent en sortie un vecteur de 300 unités. Pour chaque mot, le vecteur des valeurs numériques produites par la couche cachée sera utilisé comme embeddings combiné (nommés ***GTW-D/GTW-O***).

La différence entre l'auto-encodeur ordinaire et de débruitage vient de la notion de corruption aléatoire des entrées au cours de l'apprentissage dans le dernier cas. La corruption consiste à initialiser une partie des données à zéro. Cette corruption rend l'auto-encodeur plus généraliste en découvrant des paramètres plus robustes qu'un auto-encodeur ordinaire. À notre connaissance, les auto-encodeurs ne sont pas utilisés pour la combinaison, mais juste utilisés pour apprendre une représentation compressée pour un ensemble de données, généralement dans le but de réduire le nombre de dimensions.

4 Système de détection d'erreurs

Le système de détection d'erreurs doit attribuer une étiquette *correcte* ou *erreur* à chaque mot en se basant sur un ensemble de paramètres. Cette attribution est faite en analysant chaque mot dans son contexte.

4.1 Paramètres utilisés

Dans cette section, nous décrivons les paramètres recueillis pour chaque mot et comment ceux-ci sont extraits. Certains de ces paramètres sont identiques à ceux présentés dans (Béchet & Favre, 2013).

Chaque mot est représenté par un vecteur composé des paramètres suivants :

Mesures de confiance du SRAP : il s'agit des probabilités *a posteriori* (PAP) générées par le SRAP. La PAP est calculée à partir du réseau de confusion qui est approximée par la somme des probabilités *a posteriori* de toutes les transitions passant par ce mot et qui sont en concurrence avec lui.

Paramètres lexicaux : ils se composent de la longueur du mot (nombre de lettres) et trois indices binaires indiquant si les trois 3-grammes contenant le mot courant ont été vus dans le corpus d'apprentissage du modèle de langue du SRAP.

Paramètres syntaxiques : ils se composent de l'étiquette syntaxique (POS) et du gouverneur du mot courant ainsi que des liens de dépendances entre le mot courant et son gouverneur. [2]

Paramètres prosodiques : il s'agit de deux ensembles de paramètres. Les premiers sont extraits à partir de l'alignement forcé des transcriptions avec le signal audio : nombre de phonèmes, durée moyenne des phonèmes et durée de la pause précédent le mot. Le dernier paramètre prosodique correspond à la fréquence F0 [3]. Ces paramètres sont détaillés dans (Ghannay *et al.*, 2015b).

Mot : il s'agit de la représentation orthographique du mot dans l'étiqueteur de séquence à base de CRF et de son embedding dans le système neuronal.

4.2 Architecture

Nous utilisons une architecture neuronale basée sur une stratégie multi-flux pour l'apprentissage du réseau de neurones, nommée Perceptron Multicouche MultiStream (*MLP-MS*). Une description détaillée de cette architecture est présentée dans (Ghannay *et al.*, 2015a). Cette architecture illustrée dans la figure 1 est utilisée afin de mieux intégrer des informations contextuelles à partir des mots voisins.

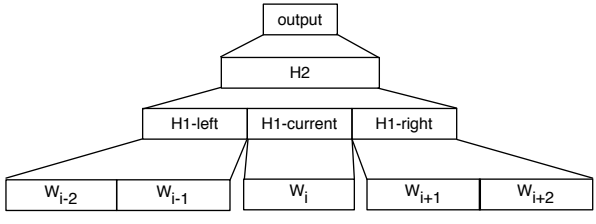


FIGURE 1: L'architecture MLP-MS pour la détection des erreurs de SRAP.

5 Expériences et résultats

5.1 Données expérimentales

Les données expérimentales sont issues du corpus français ETAPE (Gravier *et al.*, 2012), composé d'enregistrements audio d'émissions télévisées (Broadcast News) et de leurs transcriptions manuelles. Ce corpus est enrichi avec des transcriptions automatiques générées par le système *LIUM SRAP*. Il s'agit d'un système de transcription multi-passes basé sur le décodeur CMU Sphinx, utilisant des modèles acoustiques GMM/HMM. Ce système a gagné la campagne d'évaluation ETAPE en 2012. Une description détaillée est présentée dans (Deléglise *et al.*, 2009).

2. Ces paramètres sont fournis par la boite à outils MACAON http ://macaon.lif.univ-mrs.fr
3. La F0 est obtenue en analysant le signal avec la boite à outils Praat [4]

Les transcriptions automatiques ont été alignées avec les transcriptions de référence en utilisant l'outil *sclite* [5]. À partir de cet alignement, chaque mot dans le corpus a été étiqueté *correct* ou *erreur*. La description des données expérimentales est présentée dans le tableau 1.

Nom	#mots ref	#mots hyp	WER
Train	349K	316K	25.3
Dev	54K	50K	24.6
Test	58K	53K	21.9

TABLE 1: Description des données expérimentales.

5.2 Résultats expérimentaux

Cette section présente les résultats expérimentaux de notre système de détection d'erreurs *MLP-MS* et les compare à un système état de l'art basé sur les CRFs implémenté avec *Wapiti* [6]. Les résultats sont évalués en termes de rappel (R), précision (P) et F-mesure (F) pour la détection de mots erronés et le taux d'erreur de classification globale (CER). La mesure de confiance calibrée produite par les systèmes de détection d'erreurs est évaluée en terme d'Entropie Croisée Normalisée (NCE). Enfin, la significativité de nos résultats est mesurée en utilisant un intervalle de confiance à 95 %.

Afin de mesurer plus particulièrement l'apport des paramètres prosodiques, l'ensemble des paramètres présentés en section 4.1 exceptés les paramètres prosodiques sont utilisés en section 5.2.1 puis la section 5.2.2 présente les résultats lorsque tous les paramètres sont utilisés.

5.2.1 Performance des différents embeddings de mots dans le système neuronal

Un ensemble d'expériences est effectué afin d'évaluer l'impact des différents types d'embeddings ainsi que celui de leurs combinaisons. Les systèmes de détection d'erreurs (MLP-MS et CRF) sont entrainés sur le corpus d'apprentissage *Train* et optimisés sur le corpus de développement *Dev*. Les résultats expérimentaux présentés dans le tableau 2 montrent que notre proposition de combiner les embeddings est utile et améliore significativement les résultats en termes de *CER* par rapport à l'utilisation d'embeddings non-combinés. Les meilleures combinaisons *GTW-D200* et *GTW-O200* conduisent à une réduction du CER comprise entre 5 % et 5,3 % par rapport à l'approche CRF. Ces deux embeddings combinés sont utilisés dans la suite des expériences.

		Label erreur			Globale
Approches	**Représentation**	**P**	**R**	**F**	**CER**
Neuronale	glove	67.80	53.23	59.64	10.60
	tur	70.63	48.61	57.58	10.54
	w2v	72.25	46.65	56.69	10.49
	GTW 300	69.68	52.23	59.71	10.38
	GTW-PCA200	71.82	47.37	57.09	10.48
	GTW-O200	71.78	54.35	61.86	**9.86**
	GTW-D200	69.61	58.24	63.42	**9.89**
CRF	discrète	69.67	51.89	59.48	10.41

TABLE 2: Comparaison de différents types d'embeddings dans *MLP-MS* sur le corpus *Dev*

5. http ://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm
6. http ://wapiti.limsi.fr

Le tableau 3 compare les performances des embeddings *GTW-D200* et *GTW-O200* à celles du système CRF sur le corpus de test. Ces représentations réalisent respectivement 6.14% et 7.84% de réduction de CER par rapport aux CRF.

		Label erreur			Globale	
Corpus	**Approche**	**P**	**R**	**F**	**CER**	**95% intervalle de confiance**
Test	CRF	68.34	49.66	57.52	8.79	[8.55 ; 9.04]
	GTW-O200	**71.00**	54.76	61.83	**8.10**	[7.87 ; 8.33]
	GTW-D200	68.23	**58.35**	**62.90**	8.25	[8.01 ; 8.49]

TABLE 3: Performance des meilleurs embeddings sur le corpus *Test*.

5.2.2 Performance des paramètres prosodiques

Le tableau 4 présente l'impact des paramètres prosodiques sur les résultats présentés ci-dessus. L'ajout de ces paramètres conduit à une réduction du CER par rapport aux résultats des tableaux 2 et 3. De plus, nos systèmes *GTW-D200+PROS* et *GTW-O200+PROS* obtiennent des améliorations significatives par rapport aux CRF respectivement de 8,65 % et 9,45 % de réduction en CER.

		Label erreur			Globale	
Corpus	**Approche**	**P**	**R**	**F**	**CER**	**95% confiance interval**
Dev	CRF+PROS	69.89	52.86	60.20	10.29	[10.03 ; 10.56]
	GTW-O200+PROS	69.76	60.50	64.80	9.67	[9.40 ; 9.93]
	GTW-D200+PROS	**70.40**	**60.57**	**65.11**	**9.55**	[9.30 ; 9.81]
Test	CRF+PROS	68.95	51.82	59.17	8.57	[8.33 ; 8.81]
	GTW-O200+PROS	**69.01**	**60.95**	**64.73**	**7.96**	[7.73 ; 8.20]
	GTW-D200+PROS	68.68	60.65	64.42	8.03	[7.80 ; 8.27]

TABLE 4: Performance des paramètres prosodiques sur les corpus *Dev* et *Test*.

5.2.3 Calibration des mesures de confiance

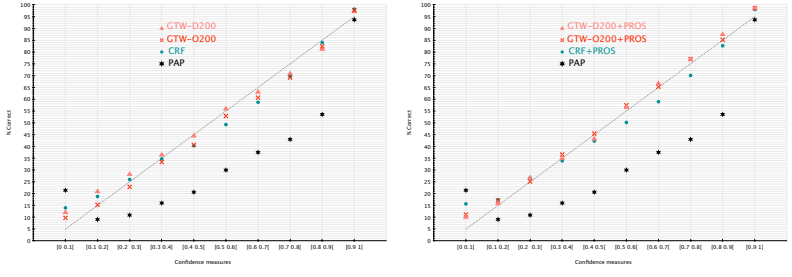
Dans une volonté d'améliorer la qualité des mesures de confiance, le problème de leur calibration a été abordé dans (Yu *et al.*, 2011). Cette étape de post-traitement est considérée comme une technique d'adaptation spéciale appliquée à la mesure de confiance afin de prendre des décisions optimales. L'utilisation de réseaux de neurones artificiels est l'une des méthodes proposées pour cette étape de post-traitement.

D'après les scores NCE présentés dans le tableau 5, nous démontrons la validité de notre approche pour produire une mesure de confiance bien calibrée, tandis que la probabilité *a posteriori* fournie par le système LIUM SRAP n'est pas calibrée. Le système CRF produit également une mesure de confiance bien calibrée. En outre, l'utilisation des caractéristiques prosodiques améliore les scores des NCE pour tous les systèmes.

Comme le montre la figure 2, les probabilités dérivées de nos systèmes neuronaux et CRF correspondent à la probabilité des mots corrects. En outre, les courbes sont bien alignées avec la diagonale, en particulier pour nos systèmes neuronaux avec des paramètres prosodiques.

Name	PAP	proba softmax GTW-D200	proba softmax GTW-O200	CRF
sans paramètres prosodiques				
Dev	-0.064	0.425	0.443	**0.445**
Tes	-0.044	0.448	**0.461**	0.457
avec paramètres prosodiques				
Dev	-0.064	0.461	**0.463**	0.449
Test	-0.044	0.471	**0.477**	0.463

TABLE 5: score NCE pour la PAP et les mesures de confiances issues des systèmes MLP-MS et CRF.



(a) (b)

FIGURE 2: Pourcentage de mots corrects en fonction des scores de la PAP et des mesures de confiances issues des systèmes MLP-MS et CRF sans (a) et avec paramètres prosodiques (b).

6 Conclusion

Dans cet article, nous avons évalué l'intégration de différentes représentations continues de mot (embeddings) sur la tâche de détection d'erreurs de SRAP. La partie expérimentale, effectuée sur le corpus ETAPE, a montré la validité de notre approche pour la détection des erreurs. Nous avons notamment proposé différents types d'embeddings simples et combinés et montré le gain obtenu par l'utilisation des embeddings combinés. De plus, nous avons prouvé l'apport significatif de l'utilisation des paramètres prosodiques, en plus de ceux syntaxiques et lexicaux classiques. En outre, les résultats obtenus sont meilleurs que ceux d'un système état de l'art basé sur les champs aléatoires conditionnels. Pour terminer, nous nous sommes attachés à démontrer que les mesures de confiances produites par notre système sont bien calibrées.

Remerciements

Ce travail a été partiellement financé par la commission européenne à travers le projet EUMSSI, sous le numéro de contrat 611 057, dans le cadre de l'appel FP7-ICT-2013-10. Ce travail a également été partiellement financé par l'Agence nationale française de recherche (ANR) à travers le projet VERA, sous le numéro de contrat ANR-12-BS02-006-01.

Références

BÉCHET F. & FAVRE B. (2013). ASR error segment localisation for spoken recovery strategy. In *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference*.

COLLOBERT R., WESTON J., BOTTOU L., KARLEN M., KAVUKCUOGLU K. & KUKSA P. (2011). Natural Language Processing (Almost) from Scratch. volume 12, p. 2493–2537 : JMLR.

DELÉGLISE P., ESTÈVE Y., MEIGNIER S. & MERLIN T. (2009). Improvements to the LIUM French ASR system based on CMU Sphinx : what helps to significantly reduce the word error rate ? In *Interspeech*, Brighton, UK.

GHANNAY S., ESTÈVE Y. & CAMELIN N. (2015a). Word embeddings combination and neural networks for robustness in asr error detection. In *European Signal Processing Conference (EUSIPCO 2015)*, Nice (France).

GHANNAY S., ESTÈVE Y., CAMELIN N., DUTREY C., SANTIAGO F. & ADDA-DECKER M. (2015b). Combining continuous word representation and prosodic features for asr error prediction. In A.-H. DEDIU, C. MARTÍN-VIDE & K. VICSI, Eds., *Statistical Language and Speech Processing*, volume 9449 of *Lecture Notes in Computer Science*, p. 84–95. Springer International Publishing.

GOLDWATER S., JURAFSKY D. & MANNING C. D. (2010). Which words are hard to recognize ? prosodic, lexical, and disfluency factors that increase speech recognition error rates. *Speech Communication*, p. 181–200.

GRAVIER G., ADDA G., PAULSSON N., CARRÉ M., GIRAUDEL A. & GALIBERT O. (2012). 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)*.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

PARADA C., DREDZE M., FILIMONOV D. & JELINEK F. (2010). Contextual information improves OOV detection in speech. In *Human Language Technologies : Proceddings of the North American Chapter of the Association for Computational Linguistics (NAACL'10)*.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). Glove : Global vectors for word representation. In *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, volume 12.

SCHWENK H., DCHELOTTE D. & GAUVAIN J.-L. (2006). Continuous space language models for statistical machine translation. In *Proceedings of COLING/ACL*, COLING-ACL '06, p. 723–730, Stroudsburg, PA, USA : Association for Computational Linguistics.

STOYANCHEV S., SALLETMAYR P., YANG J. & HIRSCHBERG J. (2012). Localized detection of speech recognition errors. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, p. 25–30.

TURIAN J., RATINOV L. & BENGIO Y. (2010). Word representations : A simple and general method for semisupervised learning. p. 384–394.

YIK-CHEUNG T., LEI Y., ZHENG J. & WANG W. (2014). ASR error detection using recurrent neural network language model and complementary ASR. In *Proceddings of Acoustics, Speech and Signal Processing (ICASSP 2014)*, p. 2312–2316.

YU D., LI J. & DENG L. (2011). Calibration of confidence measures in speech recognition. In *IEEE Transactions on Audio, Speech, and Language Processing*, volume 19, p. 2461–2473.