XXXIIe Journées d'Études sur la Parole
4-8 June 2018, Aix-en-Provence, France

Représentations de phrases dans un espace continu spécifiques à la tâche de détection d'erreurs.

Sahar Ghannay Nathalie Camelin Yannick Estève
LIUM, Le Mans Université, France
firstname.lastname@univ-lemans.fr

RÉSUMÉ

Cet article présente une étude sur la modélisation des erreurs de reconnaissance de la parole au niveau de la phrase, afin de compenser certains phénomènes mis en avant par l'analyse des sorties du système de détection d'erreurs que nous avons précédemment proposé. Nous avons étudié trois approches différentes, qui sont fondées respectivement sur l'utilisation des représentations continues (*embeddings*) de phrases dédiées à la tâche de détection d'erreur, d'un modèle contextuel probabiliste (MCP) et d'un réseau de neurones récurrent BLSTM. Une approche pour construire les *embeddings* spécifiques à la tâche est proposée et comparée à l'approche Doc2vec. Les expériences sont effectuées sur des transcriptions automatiques du corpus ETAPE générées par le système de reconnaissance automatique du LIUM. Elles montrent que les *embeddings* spécifiques à la tâche obtiennent de meilleurs résultats que les *embeddings* génériques et que leur intégration dans notre système améliore les résultats par rapport aux MCP et BLSTM.

ABSTRACT

Task specific sentence embeddings for ASR error detection

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 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 (PCM) and 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 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 PCM and BLSTM models.

MOTS-CLÉS : détection d'erreur, reconnaissance de la parole, réseau de neurones, représentation continue de phrase.

KEYWORDS: error detection, speech recognition, neural network, sentence embeddings.

1 Introduction

Les récentes avancées scientifiques dans le domaine du traitement automatique de la parole ainsi que la disponibilité de dispositifs de calcul puissants, ont conduit à l'obtention de performances

10.21437/JEP.2018-72

acceptables d'un point de vue applicatif dans le domaine de la reconnaissance automatique de la parole (RAP). Cependant, malgré ces performances, les sytèmes de RAP (SRAP) génèrent encore des erreurs de mots dans les transcriptions automatiques. Cela s'explique notamment par leurs sensibilités à la variabilité : d'environnement acoustique, de locuteur, de style de langage, de la thématique du discours, *etc*. Ces erreurs présentent un obstacle à l'application de certains traitements automatiques tels que l'extraction d'information, la traduction de la parole, la compréhension de la parole,*etc*.

Depuis deux décennies, de nombreuses études se focalisent sur la détection des erreurs de SRAP. Habituellement, les meilleurs systèmes de détection d'erreurs sont fondés sur l'utilisation des champs aléatoires conditionnels (CRF) comme dans (Parada *et al.*, 2010) et (Béchet & Favre, 2013). Des travaux récents ont commencé à appliquer les réseaux de neurones pour la tâche de détection d'erreurs. Dans ces travaux (Tam *et al.*, 2014; Ogawa & Hori, 2017), différentes architectures neuronales ont été exploitées : perceptrons multi-couches, réseaux de neurones récurrents, *etc.* Dans nos études précédentes (Ghannay *et al.*, 2015c, 2016a,b), nous avons étudié l'utilisation de différents types d'*embeddings* de mot. Dans (Ghannay *et al.*, 2015c), nous avons proposé une approche neuronale pour la détection d'erreurs dans les transcriptions automatiques et pour la calibration des mesures de confiance issues d'un SRAP. Nous avons également étudié la combinaison de différents types d'*embeddings* afin de tirer profit de leurs complémentarités. Le système de détection d'erreurs proposé inclut comme sources d'information : les *embeddings* de mots linguistiques, les descripteurs syntaxiques, lexicaux et prosodiques ainsi que des informations contextuelles extraites des mots voisins. Nous avons également enrichi notre système par des *embeddings* acoustiques de mots. L'utilisation de ces derniers en plus des autres descripteurs a amélioré les performances du système de détection d'erreurs proposé (Ghannay *et al.*, 2016a).

Dans cet article, nous présentons tout d'abord un résumé de nos études précédentes. Nous rappelons les performances obtenues par notre système de détection d'erreurs ainsi qu'une partie des résultats de l'étude sur l'analyse d'erreurs de ce système. Ensuite, pour compenser certains phénomènes mis en avant par cette analyse, nous proposons une étude sur la modélisation de l'erreur de reconnaissance au niveau de la phrase. Nous avons étudié trois approches différentes, fondées respectivement sur : l'utilisation des *embeddings* de phrases dédiées à la tâche de détection d'erreurs, un modèle contextuel probabiliste (MCP), et un réseau de neurones récurrent BLSTM (*bidirectional long short term memory*). Une approche pour construire les *embeddings* de phrases spécifiques à la tâche de détection d'erreurs est proposée et comparée à l'approche Doc2vec.

2 Système de détection d'erreurs

Le système de détection d'erreurs s'appuie sur une architecture neuronale fondée sur une stratégie multi-flux pour l'apprentissage d'un réseau de neurones, nommée Perceptron Multicouche Multi-Stream (*MLP-MS*).Une description détaillée de cette architecture est présentée dans (Ghannay *et al.*, 2015b).

2.1 Ensemble de descripteurs

Le système *MLP-MS* doit attribuer une étiquette *correct* ou *erreur* à chaque mot en l'analysant dans son contexte. Cette attribution est faite en s'appuyant sur l'ensemble de descripteurs suivants, dont certains sont identiques à ceux présentés dans (Béchet & Favre, 2013), pour chaque mot :

— Probabilités *a posteriori* générées par le SRAP.
— Descripteurs lexicaux extraits des sorties de SRAP : 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.
— Descripteurs syntaxiques fournis par la boite à outils MACAON [1] appliquée aux sorties de SRAP. Des analyseurs morphosyntaxiques et de dépendances sont utilisés pour extraire les étiquettes syntaxiques, le gouverneur du mot courant et les liens de dépendance entre le mot courant et son gouverneur.
— Descripteurs prosodiques : le nombre de phonèmes, la durée moyenne des phonèmes, la durée de la pause précédant le mot sont extraits à partir de l'alignement forcé des transcriptions avec le signal audio. Ces paramètres sont détaillés dans (Ghannay *et al.*, 2015c).
— Le mot. Dans MLP-MS, il est représenté par son *embeddding* linguistique qui correspond à la combinaison par auto-encodeur de trois *embedddings* différents : *w2vf-deps* (Levy & Goldberg, 2014), *skip-gram* fourni par *word2vec* (Mikolov *et al.*, 2013), et *GloVe* (Pennington *et al.*, 2014). Cette combinaison est décrite dans (Ghannay *et al.*, 2015c). La représentation orthographique du mot est utilisée dans le système à base de CRF (Béchet & Favre, 2013).
— Les *embeddding* acoustiques. Ils correspondent aux *embedddings* acoustiques de signal et acoustiques de mot décrits dans (Ghannay *et al.*, 2016a).

2.2 Expériences et résultats

2.2.1 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*, qui est un système 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).

Les transcriptions automatiques ont été alignées avec les transcriptions de référence en utilisant l'outil *sclite* [2]. 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 – Nombre de mots de référence (*#mots ref*), nombre de mots générés par le SRAP LIUM (*#mots hyp*) et taux d'erreur mot (*WER*) de chacun des sous-corpus issu d'ETAPE.

1. http ://macaon.lif.univ-mrs.fr
2. http ://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm

2.2.2 Résultats expérimentaux

Performance du système MLP-MS

Cette section présente les résultats expérimentaux de notre système de détection d'erreurs *MLP-MS* ainsi que l'analyse de ses sorties en fonction de l'empan moyen de l'erreurs, et les compare à un système état de l'art basé sur les CRFs implémentés avec *Wapiti* [3]. Les résultats sont évalués en termes de rappel (R), précision (P) et F-mesure (F) pour la détection de mots erronés et de taux d'erreur de classification globale (CER).

Corpus	**Approche**	**Etiquette *erreur***			**Global**
		P	**R**	**F**	**CER**
Dev	CRF	0,70	0,55	0,62	10,12
	MLP-MS	0,72	0,60	0,65	**9,38**
Test	CRF	0,69	0,54	0,61	8,46
	MLP-MS	**0,70**	0,61	0,65	**7,75**

TABLE 2 – Comparaison des performances des systèmes MLP-MS et CRF.

Les résultats expérimentaux présentés dans le tableau 2, montrent que notre système MLP-MS obtient de meilleurs résultats sur la détection de l'étiquette *erreur* et améliore significativement [4] les résultats par rapport à l'approche CRF.

Empan moyen d'erreurs

Dans cette section, nous nous intéressons à l'analyse des sorties de notre système de détection d'erreurs MLP-MS afin de percevoir les erreurs de reconnaissance qui sont difficiles à détecter. Ces analyses sont réalisées sur le corpus Dev en fonction de l'empan moyen de l'erreurs, *i.e.* une suite contiguë de mots erronés (Ghannay *et al.*, 2015a). Le tableau 3 présente la taille moyenne (en nombre de mots) des empans et l'écart type pour la vérité terrain, les prédictions et les prédictions correctes de deux systèmes : MLP-MS et CRF.

Corpus	**Approche**	**Provenance du mot**	**Taille moyenne de l'empan**	**Écart type**
Train		vérité terrain	3,03	1,72
Dev			3,24	2,15
Dev	CRF	prédictions	3,28	1,77
		prédictions correctes	2,88	1,34
	MLP-MS	prédictions	2,82	1,28
		prédictions correctes	2,66	1,05

TABLE 3 – Empan moyen et écart-type pour la vérité terrain, les prédictions et les prédictions correctes de MLP-MS et CRF

3. http ://wapiti.limsi.fr
4. Un intervalle de confiance à 95% a été calculé et les résultats significativement meilleurs ont été soulignés.

On observe que la taille moyenne de l'empan des prédictions du CRF est proche de celle de la vérité terrain. En revanche, celle du système MLP-MS est, elle, plus petite de $12,9\%$ par rapport à la vérité terrain avec un écart type plus petit de $40,5\%$. Les prédictions correctes, tant pour celles produites par le système CRF que par celui proposé, présentent des empans de taille bien inférieure à la vérité terrain.

Nous supposons que l'écart lié à la taille de l'empan d'erreur entre la vérité terrain, les prédictions et les prédictions correctes, est dû à l'architecture de MLP-MS. Contrairement aux CRF, dont le décodeur produit des séquences d'étiquettes à partir d'un calcul sur l'ensemble de la séquence d'entrée, notre système prend ses décisions en s'appuyant sur un contexte local restreint.

S'appuyant sur cette observation, nous avons choisi d'explorer trois approches différentes détaillées dans ce qui suit.

3 Intégration d'informations globales à la phrase pour la détection d'erreurs

3.1 Représentation de la phrase dans un espace continu

Dans cette section, nous proposons d'enrichir notre système MLP-MS par des informations caractérisant la phrase, en exploitant des *embeddings* de phrases. Ces derniers ont été utilisés avec succès dans les tâches de classification de phrases et l'analyse de sentiments (Le & Mikolov, 2014; Tang *et al.*, 2016). Ces représentations peuvent être apprises d'une manière générale en utilisant l'outil *Doc2vec* (Le & Mikolov, 2014), ou d'une manière spécifique à la tâche, comme dans (Ren *et al.*, 2016).

3.1.1 Embeddings généralistes

Ce premier type *d'embeddings* de phrases s'appuie sur la méthode de sacs de mots distribués DBOW (*Distributed bag of words*) fournie par *Doc2vec* (Le & Mikolov, 2014). L'architecture DBOW consiste à prédire des mots choisis aléatoirement en fonction du paragraphe auquel ils appartiennent. Elle est apprise sur le corpus ETAPE pour construire un *embedding* de 100 dimensions pour chaque transcription automatique, nommé Em_{DBOW}.

3.1.2 Embeddings spécifiques à la tâche

Les Em_{DBOW} portent une information sur la sémantique contenue dans les transcriptions, mais ne portent probablement pas d'information sur les erreurs de transcriptions. C'est pourquoi nous proposons de construire des *embeddings* de phrases spécifiques à la tâche de détection d'erreurs. Pour ce faire, nous proposons d'utiliser les *embeddings* extraits d'un réseau de neurones convolutif (CNN) appris pour la classification des transcriptions automatiques en deux catégories de phrases : *peu erronées* (PE) ou *très erronées* (TE). Les *embeddings* extraits du *CNN* sont ainsi susceptibles de capter des informations sur les erreurs.

Le CNN est appris sur les transcriptions d'ETAPE annotées en *peu erronées* ou *très erronées*. En effet, nous avons considéré arbitrairement une phrase comme très erronée si 20% des mots qui la composent sont incorrects. Les phrases comprenant moins de 20% de mots incorrects sont alors considérées comme peu erronées (ensemble incluant les phrases totalement correctes). Le CNN prend en entrée le vecteur de descripteurs de la phrase et attribue en sortie une étiquette *PE* ou *TE* globale à la phrase. Le vecteur de descripteurs correspond à la concaténation des vecteurs de descripteurs des mots qui la composent, décrits dans la section 2.1. Le CNN est composé de deux couches de convolution et de sous-échantillonnage, suivies par deux couches de neurones qui sont totalement connectées sous la forme d'un MLP. La couche juste avant la couche de sortie *Softmax* est utilisée comme *embedding* de phrases de 100 dimensions, nommé Em_{CNN}. Le CNN obtient 13,5% de taux d'erreur de classification des transcriptions sur Test.

3.1.3 Résultats

Nous résumons dans la table 4 les performances des *embeddings* de phrases Em_{DBOW} et Em_{CNN}. Elles sont comparées à celles obtenues par le système MLP-MS sans *embedding* de phrase (table 2).

Corpus	**Représentation de phrase**	**Étiquette *erreur***			**Global**
		P	**R**	**F**	**CER**
Dev	Em_{DBOW}	**0,73**	0,58	0,65	9,36
	Em_{CNN}	0,72	0,60	0,66	**9,26**
Test	Em_{DBOW}	**0,72**	0,57	0,60	7,72
	Em_{CNN}	**0,72**	0,58	0,64	**7,69**

TABLE 4 – Performances des *embeddings* de phrases Em_{DBOW} et Em_{CNN} sur Dev et Test

On remarque que les deux types d'*embeddings* de phrases apportent une légère amélioration par rapport aux résultats de MLP-MS. L'*embedding* Em_{CNN} obtient de meilleurs résultats que l'*embedding* Em_{DBOW} avec 1,27% et 0,77% de réduction de CER par rapport aux résultats de MLP-MS, respectivement sur Dev et Test. Nous pouvons émettre l'hypothèse que les *embeddings* extraits du CNN ont capté une information utile sur l'erreur.

Ce système sera nommé désormais MLP-MS$_{Em_{CNN}}$.

3.2 Modèle contextuel probabiliste pour une décision globale

Une autre approche pour compenser les lacunes remarquées lors de notre analyse d'erreurs consiste à utiliser un modèle contextuel probabiliste (MCP) qui porte des informations sur la distribution d'erreurs. Nous espérons ici corriger le problème de la taille de l'empan d'erreurs mal capturée par notre système de détection.

Cette approche est similaire à celle utilisée par (Dufour *et al.*, 2014) pour la détection automatique de segments de parole spontanée dans des émissions télévisées. Les auteurs ont proposé d'étendre un processus de classification locale à l'aide d'un modèle contextuel probabiliste d'étiquetage de séquences qui prend en compte l'étiquetage (parole préparée *vs.* parole spontanée) des segments voisins dans une fenêtre de taille 3. Grâce à cette extension, l'étiquetage, qui était issu d'une succession de décisions locales, devient un processus global.

Nous proposons d'appliquer cette idée à notre approche pour la détection d'erreur. Jusqu'à présent, l'étiquetage en *erreur vs. correct* des transcriptions automatiques par notre approche neuronale consistait en autant de classifications indépendantes que de mots à étiqueter. En tenant compte des classifications locales des mots voisins dans une fenêtre contextuelle de taille 5 identique à celle de l'entrée de notre système de détection d'erreurs, nous espérons lisser au niveau de la phrase le résultats de ces classifications. Pour cela, un modèle probabiliste d'ordre n de distribution d'erreurs est utilisé : ce modèle estime la probabilité que le mot courant soit erroné en fonction de la justesse des 4 mots qui l'entourent.

3.2.1 Résultats

Nous avons utilisé la boite à outils *OpenFst* [5] pour créer le modèle sur les transcriptions automatiques du corpus ETAPE et les sorties de deux systèmes de détection : le système de base MLP-MS et le système MLP-MS$_{Em_{CNN}}$ qui intègre des connaissances sur la phrase.

Les systèmes résultants sont nommés avec l'extension -MCP. Les résultats obtenus par cette approche pour la détection d'erreurs sont résumés dans la table 5.

Corpus	**Approche**	**Étiquette *erreur***			**Global**
		P	**R**	**F**	**CER**
Dev	MLP-MS-MCP	**0,73**	0,58	0,65	9,31
	MLP-MS$_{Em_{CNN}}$-MCP	0,73	0,60	0,65	**9,23**
Test	MLP-MS-MCP	0,72	0,59	0,65	**7,67**
	MLP-MS$_{Em_{CNN}}$-MCP	0,73	0,57	0,64	7,69

TABLE 5 – Performances du modèle contextuel probabiliste pour la détection d'erreurs.

On observe que l'application du MCP aux sorties du système MLP-MS permet une légère diminution du CER tant sur Dev que sur Test. celui-ci est comparable à celui obtenu par le système MLP-MS$_{Em_{CNN}}$ qui intègre des informations globales sur la phrase.

L'application du MCP aux sorties de MLP-MS$_{Em_{CNN}}$ n'a amélioré que légèrement les résultats sur Dev. Cela peut s'expliquer par le fait que ce système intègre déjà des connaissances sur la phrase, et l'information apportée par l'approche globale est redondante.

3.3 Réseau de neurones BLSTM

Récemment, certaines architectures neuronales se sont révélées efficaces pour faire le traitement des séquences (Sutskever *et al.*, 2014). Il est donc intéressant de comparer l'approche neuronale utilisée jusqu'à présent dans nos expériences avec l'utilisation d'une architecture BLSTM. Cette comparaison nous permet d'évaluer l'impact des représentations continues de phrase dans une architecture classique par rapport à l'utilisation d'une architecture conçue pour apprendre des informations contextuelles distantes. Ce type d'architecture a notamment été utilisé avec succès pour la tâche de détection d'erreurs de transcription dans (Ogawa & Hori, 2017).

5. http ://www.openfst.org/twiki/bin/view/FST/WebHome

Dans nos expériences, le BLSTM est composé de deux couches de 512 unités chacune : 256 unités dans chaque direction. Il intègre les descripteurs décrits dans la section 2.1. Les résultats sont résumés dans la table 6.

Corpus	**Système**	**Étiquette *Erreur***			**Global**
		P	**R**	**F**	**CER**
Dev	BLSTM	0,70	0,64	0,67	9,28
Test	BLSTM	0,69	0,63	0,66	7,83

TABLE 6 – Comparaison de l'architecture MLP-MS proposée à l'architecture BLSTM.

Lorsque l'on compare les résultats obtenus par MLP-MS (table 2) et les résultats du BLSTM, on remarque que ceux-ci sont comparables. Le système MLP-MS$_{Em_{CNN}}$ montre des performances légèrement meilleures que celles du BLSTM. Cela nous permet ainsi de confirmer notre hypothèse sur l'utilité de l'intégration des informations globales sur la phrase dans notre système MLP-MS afin d'améliorer une prise de décision locale.

4 Conclusion

Dans cet article nous avons présenté une étude sur la modélisation des erreurs de reconnaissance de la parole au niveau de la phrase, afin de compenser certains phénomènes mis en avant par l'analyse des sorties du système de détection d'erreurs que nous avons précédemment proposé. Nous avons étudié trois approches différentes, qui sont fondées respectivement sur l'utilisation des *embeddings* de phrases dédiées à la tâche de détection d'erreurs, d'un modèle contextuel probabiliste et d'un réseau de neurones récurrent BLSTM. Nous avons également proposé une approche pour construire les *embeddings* spécifiques à la tâche et les comparer à l'approche Doc2vec. Les expériences sont effectuées sur des transcriptions automatiques du corpus ETAPE générées par le système de reconnaissance automatique du LIUM. Elles montrent que les *embeddings* de phrase spécifiques à la tâche obtiennent de meilleurs résultats que les *embeddings* génériques. De plus, leur intégration dans notre système améliore les résultats par rapport à l'application du modèle contextuel probabiliste sur les sorties du système MLP-MS et également par rapport à l'utilisation d'un BLSTM.

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 localization for spoken recovery strategy. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, p. 6837–6841.

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.

DUFOUR R., ESTÈVE Y. & DELÉGLISE P. (2014). Characterizing and detecting spontaneous speech : Application to speaker role recognition. *Speech Communication*, **56**, 1–18.

GHANNAY S., CAMELIN N. & ESTÈVE Y. (2015a). Which asr errors are hard to detect ? In *Workshop Errors by Humans and Machines in multimedia, multimodal and multilingual data processing (ERRARE 2015)*, Sinaia (Romania).

GHANNAY S., ESTÈVE Y. & CAMELIN N. (2015b). 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. & DELEGLISE P. (2016a). Acoustic word embeddings for asr error detection. In *Interspeech 2016*, San Francisco (CA, USA).

GHANNAY S., ESTEVE Y., CAMELIN N. & DELÉGLISE P. (2016b). Evaluation of acoustic word embeddings. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, p. 62–66.

GHANNAY S., ESTÈVE Y., CAMELIN N., DUTREY C., SANTIAGO F. & ADDA-DECKER M. (2015c). 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).

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)*.

LE Q. V. & MIKOLOV T. (2014). Distributed representations of sentences and documents. In *ICML*, volume 14, p. 1188–1196.

LEVY O. & GOLDBERG Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, p. 302–308.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at ICLR*.

OGAWA A. & HORI T. (2017). Error detection and accuracy estimation in automatic speech recognition using deep bidirectional recurrent neural networks. *Speech Communication*, **89**, 70–83.

PARADA C., DREDZE M., FILIMONOV D. & JELINEK F. (2010). 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*.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, volume 14, p. 1532–1543.

REN Y., WANG R. & JI D. (2016). A topic-enhanced word embedding for twitter sentiment classification. *Information Sciences*, **369**, 188–198.

SUTSKEVER I., VINYALS O. & LE Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, p. 3104–3112.

TAM Y.-C., LEI Y., ZHENG J. & WANG W. (2014). Asr error detection using recurrent neural network language model and complementary asr. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, p. 2312–2316 : IEEE.

TANG D., WEI F., QIN B., YANG N., LIU T. & ZHOU M. (2016). Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, **28**, 496–509.