Traitement automatique des langues

Apprentissage profond pour le traitement automatique des langues

sous la direction de
Alexandre Allauzen
Hinrich Schütze

Vol. 59- n°2 / 2018

ATALA

Apprentissage profond pour le traitement automatique des langues

ATALA

Revue de
l'Association
pour le Traitement
Automatique
des Langues

TAL
Vol.
59

n°2
2018

Apprentissage profond pour le traitement automatique des langues

ATALA

Traitement automatique des langues

Revue publiée depuis 1960 par l'Association pour le Traitement Automatique des LAngues (ATALA), avec le concours du CNRS, de l'Université Paris VII et de l'Université de Provence

ISSN 1965-0906

https ://www.atala.org/revuetal

Traitement automatique des langues

Comité de rédaction

Traitement automatique des langues

Volume 59– n°2/2018

APPRENTISSAGE PROFOND POUR LE TRAITEMENT AUTOMATIQUE DES LANGUES

Table des matières

Apprentissage profond pour le traitement automatique des langues

Alexandre Allauzen* — **Hinrich Schütze****

* *LIMSI-CNRS, Université Paris-Sud et Université Paris-Saclay*
** *Center for Information and Language Processing, Ludwig-Maximilians-Universität (LMU), München*

RÉSUMÉ. *Ces dernières décennies, les réseaux de neurones artificiels et plus généralement leur apprentissage dit profond ont renouvelé les perspectives de recherche en traitement automatique des langues (TAL). Par leur capacité en termes d'apprentissage de représentations, les réseaux neuronaux ont permis des avancées importantes pour le TAL et ce pour de nombreuses tâches par exemple l'analyse syntaxique, la classification de documents, la reconnaissance automatique de la parole et la traduction automatique. Néanmoins, ces évolutions récentes posent de nombreuses questions scientifiques. Si les performances obtenues par les modèles neuronaux impressionnent souvent, les architectures déployées sont complexes à concevoir et à optimiser. La vision de ces modèles comme une boîte noire est problématique tant l'interprétation des résultats et la compréhension de ce qui est appris restent obscures. Ce numéro spécial propose donc d'explorer les apports de l'apprentissage profond pour le TAL, d'en montrer à la fois les promesses, les limites et les singularités.*

ABSTRACT. *During the last decades, artificial neural networks and deep learning approaches have strongly renewed the research perspectives in Natural Language Processing (NLP). Neural networks provide an efficient way for representation learning, yielding important improvement in several tasks. These tasks include document classification, syntactic parsing, automatic speech recognition and machine translation. Whereas the performances achieved by neural networks are impressive, their conception and optimization are still challenging. Moreover, these architectures are merely understood as efficient black boxes and their results remain difficult to interpret and explain. This special issue explores contributions of deep learning to NLP, their promises along with their limits and peculiarities.*

MOTS-CLÉS : *réseaux de neurones, apprentissage profond.*

KEYWORDS: *neural network, deep-learning.*

1. Introduction

Au cours des dernières décennies, les réseaux de neurones artificiels et plus généralement l'apprentissage profond de ces modèles (*deep-learning*) ont renouvelé les perspectives de recherche en traitement automatique des langues (TAL). Comme dans d'autres domaines, le potentiel de ces approches à modéliser des données et des tâches complexes explique leur impact. Ainsi, de nombreuses applications du TAL, avec pourtant des objectifs différents, sont concernées. Par exemple, la classification de textes implique de représenter un texte de longueur variable afin d'en extraire les caractéristiques nécessaires à la prédiction d'une classe. En traduction automatique, l'enjeu peut se schématiser par la représentation d'une phrase dans un but de génération d'une séquence de mots dans une autre langue. Cet enjeu se retrouve de manière similaire dans d'autres applications comme en reconnaissance automatique de la parole, où la séquence d'entrée est un signal acoustique, et en analyse syntaxique, où la structure à engendrer est un arbre.

Face à cette diversité des structures manipulées, de nombreux travaux en TAL ont exploré de manière souvent disjointe, d'une part, la représentation des données d'entrée (mots, phrases et documents), et d'autre part, les modèles de prédiction, par exemple effectuant la classification d'un texte ou l'inférence d'une traduction. Ainsi de nombreux travaux ont été pionniers dans l'apprentissage de représentations vectorielles pour les mots et les documents, grâce à l'analyse sémantique latente et ses variantes (Benzécri, 1981 ; Deerwester *et al.*, 1990 ; Schütze, 1992) ou des modèles de thèmes probabilistes (Hofmann, 2001 ; Blei *et al.*, 2003). Par ailleurs, l'inférence de structures a donné lieu à des modèles spécifiques comme le perceptron structuré (Collins, 2002) et les champs aléatoires conditionnels (Lafferty *et al.*, 2001) où la représentation des données d'entrée doit être spécifiée par l'utilisateur.

Une des difficultés majeures, en termes d'apprentissage automatique, est que les données langagières se caractérisent par des distributions particulières suivant la loi de Zipf (Zipf, 1935). Ces distributions sont souvent qualifiées de parcimonieuses ou de creuses et elles portent sur des unités discrètes dont l'inventaire est potentiellement grand. Ainsi la plupart des modèles utilisés par le passé ont été confrontés à des difficultés de généralisation, et les enjeux scientifiques se sont alors concentrés sur, d'une part, le développement d'estimateurs statistiques plus robustes, illustré par les nombreux travaux sur les modèles de langues et leur estimation (H. Ney, 1994 ; Chen et Goodman, 1996 ; Teh, 2006), et, d'autre part, la conception de modèles pouvant tenir compte d'une description riche des données manipulées dont les champs conditionnels aléatoires sont les représentants (Lavergne *et al.*, 2010).

2. Réseaux de neurones pour le TAL

Dès les premiers travaux, l'application des réseaux de neurones au TAL poursuit le même objectif lié à la représentation des unités linguistiques qui sont discrètes, par exemple des mots, des caractères, des catégories morphosyntaxiques ou séman-

tiques. L'objectif est de substituer à ces unités discrètes une représentation numérique continue sous forme de vecteur afin de les « plonger » dans un espace où il est possible de définir des notions de similarité qui sont donc plus propices à la généralisation. Cette notion de plongement apparaît d'abord dans le contexte des réseaux sémantiques (Hinton, 1981 ; Hinton, 1986). Puis dans (Sejnowski et Rosenberg, 1986 ; Sejnowski et Rosenberg, 1988), les auteurs introduisent les premiers plongements (ou *embedding*) de caractères pour faire de la conversion graphème-phonème et dans (Nakamura et Shikano, 1988 ; Nakamura *et al.*, 1990) les unités considérées sont des classes morphosyntaxiques.

Cette notion de plongement sera formalisée plus avant dans (Bengio *et al.*, 2003) avec cette fois-ci comme application la définition d'un modèle de langue neuronal n-grammes. L'unité considérée dans ces travaux est le mot et on parle alors de plongements lexicaux ou *word embeddings*. L'apport des réseaux de neurones réside dans leur capacité à représenter dans un espace continu les unités discrètes manipulées. Le modèle peut ainsi mieux généraliser ses connaissances extraites des données d'apprentissage en exploitant la similarité entre les unités manipulées, dans l'espace continu de représentations. La différence fondamentale avec d'autres types de représentations vectorielles comme l'analyse sémantique latente (Deerwester *et al.*, 1990) dans un contexte applicatif similaire (Bellegarda, 2000 ; Afify *et al.*, 2007) est que les représentations sont apprises conjointement avec la fonction de similarité nécessaire à l'application. Cette idée a permis des avancées rapides et notables en reconnaissance automatique de la parole (Schwenk et Gauvain, 2002) puis en traduction automatique (Schwenk *et al.*, 2006 ; Le *et al.*, 2012).

3. De l'apprentissage de représentations aux systèmes de bout en bout

Au-delà de ces premières applications, les plongements lexicaux ont par la suite été utilisés dans de nombreuses tâches du TAL. Dans (Collobert et Weston, 2008 ; Collobert *et al.*, 2011), les auteurs proposent une architecture unifiée permettant d'exploiter les plongements lexicaux pour différentes tâches d'étiquetage de séquences. Partant du constat que, dans beaucoup de langues, les textes sous format électronique sont des ressources facilement accessibles et exploitables, il est possible de pré-apprendre les plongements lexicaux sur ces données textuelles non annotées disponibles en grande quantité puis de raffiner ces représentations pour une tâche précise en utilisant cette fois les données annotées qui sont, quant à elles, disponibles en faible quantité (Collobert *et al.*, 2011 ; Mikolov *et al.*, 2013). Ce type d'approche a connu récemment un regain d'intérêt très important avec le déploiement d'architectures neuronales plus complexes et des capacités d'apprentissage bien plus importantes, tant au niveau des ressources de calcul que des données disponibles (Peters *et al.*, 2018 ; Howard et Ruder, 2018 ; Devlin *et al.*, 2018).

Les réseaux de neurones se distinguent également par leur capacité à représenter une phrase au-delà d'un simple sac de mots. Cette capacité s'appuie principalement sur deux types d'architectures qui se distinguent par les mécanismes de représenta-

tions d'une séquence, c'est-à-dire sur la façon de la décomposer ainsi que de tenir compte des dépendances et des interactions entre les mots qui la constituent. Les réseaux convolutifs sont le premier type. Inspirés par l'opérateur de convolution en traitement du signal, ces réseaux peuvent être perçus comme une généralisation des modèles n-grammes. Ils s'appuient sur des fenêtres glissantes de différentes tailles, permettant l'extraction de caractéristiques locales. Ces caractéristiques sont ensuite combinées afin de représenter la phrase dans son ensemble (Waibel *et al.*, 1990 ; Collobert et Weston, 2008 ; Kim, 2014). L'autre type d'architecture utilise les réseaux récurrents (Elman, 1990) et leurs évolutions récentes comme les réseaux LSTM, pour *Long Short Term Memory* (Hochreiter et Schmidhuber, 1997 ; Graves, 2008). Un réseau récurrent parcourt la phrase, par exemple de gauche à droite, un mot après l'autre, mettant à jour la mémoire interne du réseau à chaque pas, accumulant ainsi une vision globale de la séquence. Afin de renforcer la modélisation des dépendances à longue distance, les réseaux récurrents peuvent être bidirectionnels, parcourant la phrase de gauche à droite et de droite à gauche (Schuster et Paliwal, 1997).

L'enjeu pour ces architectures est double. D'une part, elles permettent d'apprendre à représenter les unités qui composent une séquence grâce aux plongements lexicaux, et, d'autre part, elles modélisent le mécanisme combinant ces plongements afin de représenter la séquence dans son ensemble. Ainsi, les modèles neuronaux ont dépassé le cadre de l'apprentissage de représentations pour évoluer vers des architectures de plus en plus profondes, permettant de modéliser de bout en bout des tâches d'inférence de plus en plus complexes, comme la génération d'une phrase ou d'un arbre syntaxique. Désormais, pour de nombreuses applications, les approches considérées comme état de l'art ont rapidement évolué ces dernières années, que ce soit en traduction automatique (Bahdanau *et al.*, 2014 ; Vaswani *et al.*, 2017) en reconnaissance automatique de la parole (Chan *et al.*, 2016 ; Chiu *et al.*, 2018) en synthèse vocale (van den Oord *et al.*, 2016 ; Li *et al.*, 2018), ou pour la génération de légendes d'images (Xu *et al.*, 2015). Ces systèmes, qui s'appuyaient auparavant sur différents types de modèles, sont désormais constitués d'un seul réseau de neurones.

4. Contenu du numéro

Ce numéro spécial de la revue TAL comprend les trois articles suivants :

– « *Classifying Semantic Clause Types with Recurrent Neural Networks : Analysis of Attention, Context and Genre Characteristics* », de Maria Becker, Michael Staniek, Vivi Nastase, Alexis Palmer, et Anette Frank ;

– « Prédiction de performances des systèmes de reconnaissance automatique de la parole à l'aide de réseaux de neurones convolutifs », de Zied Elloumi, Laurent Besacier, Olivier Galibert, et Benjamin Lecouteux ;

– « *Adversarial networks for machine reading* », de Quentin Grail, Julien Perez, et Tomi Silander.

Le premier article s'intéresse à la classification sémantique des propositions. Les auteurs déploient une architecture combinant des réseaux récurrents avec un mécanisme d'attention. Il explore en particulier la capacité de ce type de modèle à apprendre une représentation pertinente pour les propositions, et l'usage du mécanisme d'attention permet de mieux caractériser la nature des propriétés linguistiques apprises par le réseau. Dans le deuxième article, bien que le choix de l'architecture diffère avec la construction d'un réseau convolutif, les auteurs explorent également les propriétés apprises par le réseau afin de mieux expliquer les performances du modèle. L'application est ici de prédire les performances d'un système de reconnaissance automatique de la parole lorsqu'il est confronté à des conditions d'utilisation nouvelles. Le troisième article explore une architecture complexe qui couple deux réseaux « adversaires » afin d'apprendre un système de réponses à des questions. Ce dispositif d'apprentissage permet d'accroître la robustesse du système au bruit pouvant être présent dans les données, et de relâcher certaines contraintes de l'apprentissage supervisé.

Ces trois articles couvrent des applications différentes, avec des enjeux scientifiques bien distincts, allant de l'apprentissage de représentations aux stratégies d'apprentissage. Une thématique récurrente dans ces travaux est de montrer la capacité des réseaux neuronaux, pour des tâches complexes, d'apprendre automatiquement des caractéristiques à la fois pertinentes, en partie explicables, et robustes. Ces trois articles illustrent donc bien le potentiel et les promesses de ce type d'approche. Ils montrent aussi la complexité des solutions qui sont explorées. Ainsi la conception d'architectures neuronales pour le TAL reste un enjeu scientifique et technique qui, loin d'être une solution facile et toute prête, porte néanmoins des promesses importantes de progrès.

Remerciements

Nous tenons à remercier Sophie Rosset pour le suivi de ce numéro, les membres du comité permanent de la revue TAL, ainsi que les membres du comité spécifique à ce numéro :

– Marianna Apidianaki, LIMSI-CNRS
– Loic Barrault, Université du Maine, LIUM
– Fethi Bougares, LIUM, Université du Maine
– Marie Candito, LLF, Université Paris Diderot
– Marta R. Costa-jussà, Universitat Politecnica de Catalunya
– Benoit Crabbé, LLF, Université Paris Diderot
– Richard Dufour, LIA, Université d'Avignon
– Benoit Favre, LIS, Aix-Marseille Université
– Joseph Leroux, LIPN, Université Paris-Nord
– Matthieu Labeau, LIMSI, Université Paris-Sud

– Gwénolé Lecorvé, IRISA, Université de Rennes I, ENSSAT
– Fabrice Lefevre, LIA, Université d'Avignon
– Thomas Pellegrini, IRIT, Université de Toulouse III - Paul Sabatier
– Christian Raymond, IRISA, INSA de Rennes
– Lynda Tamine-Lechani, IRIT, Université de Toulouse III - Paul Sabatier
– Tim Van de Cruys, IRIT, CNRS

5. Bibliographie

Afify M., Siohan O., Sarikaya R., « Gaussian Mixture Language Models for Speech Recognition », *Proceedings of the IEEE international conference on Acoustics, speech, and signal processing (ICASSP)*, vol. 4, p. 29-32, April, 2007.

Bahdanau D., Cho K., Bengio Y., « Neural Machine Translation by Jointly Learning to Align and Translate », *CoRR*, 2014.

Bellegarda J. R., « Exploiting latent semantic information in statistical language modeling », *Proc. of the IEEE, Special Issue on Speech Recognition and Understanding*, vol. 88, n° 8, p. 1279-1296, 2000.

Bengio Y., Ducharme R., Vincent P., Janvin C., « A neural probabilistic language model », *Journal of Machine Learning Research*, vol. 3, p. 1137-1155, 2003.

Benzécri J.-P., *Pratique de l'analyse des données. Linguistique et lexicologie*, Dunod, 1981.

Blei D. M., Ng A. Y., Jordan M. I., « Latent dirichlet allocation », *Journal of Machine Learning Research*, vol. 3, p. 993-1022, 2003.

Chan W., Jaitly N., Le Q., Vinyals O., « Listen, attend and spell : A neural network for large vocabulary conversational speech recognition », *Proceedings of the IEEE international conference on Acoustics, speech, and signal processing (ICASSP)*, p. 4960-4964, March, 2016.

Chen S. F., Goodman J., « An Empirical Study of Smoothing Techniques for Language Modeling », *in* A. Joshi, M. Palmer (eds), *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, Morgan Kaufmann Publishers, San Francisco, p. 310-318, 1996.

Chiu C.-C., Sainath T., Wu Y., Prabhavalkar R., Nguyen P., Chen Z., Kannan A., Weiss R. J., Rao K., Gonina K., Jaitly N., Li B., Chorowski J., Bacchiani M., « State-of-the-art Speech Recognition With Sequence-to-Sequence Models », *Proceedings of the IEEE international conference on Acoustics, speech, and signal processing (ICASSP)*, 2018.

Collins M., « Discriminative training methods for hidden Markov models : theory and experiments with perceptron algorithms », *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1-8, 2002.

Collobert R., Weston J., « A unified architecture for natural language processing : deep neural networks with multitask learning », *Proceedings of the International Conference of Machine Learning (ICML)*, ACM, New York, NY, USA, p. 160-167, 2008.

Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., Kuksa P., « Natural Language Processing (Almost) from Scratch », *Journal of Machine Learning Research*, vol. 12, p. 2493-2537, 2011.

Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K., Harshman R., « Indexing by latent semantic analysis », *Journal of the American Society for Information Science*, vol. 41, n° 6, p. 391-407, 1990.

Devlin J., Chang M., Lee K., Toutanova K., « BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding », *CoRR*, 2018.

Elman J. L., « Finding structure in time », *Cognitive Science*, vol. 14, n° 2, p. 179-211, 1990.

Graves A., Supervised sequence labelling with recurrent neural networks, PhD thesis, Technical University Munich, 2008.

H. Ney U. Essen R. K., « On Structuring Probabilistic Dependences in Stochastic Language Modelling », *Computer Speech and Language*, vol. 8, n° 1, p. 1-38, 1994.

Hinton G. E., « Implementing Semantic Networks in Parallel Hardware », *in* G. E. Hinton, J. A. Anderson (eds), *Parallel Models of Associative Memory*, Erlbaum, p. 161-187, 1981.

Hinton G. E., « Learning Distributed Representations of Concepts », *Annual Conference of the Cognitive Science Society*, 1986.

Hochreiter S., Schmidhuber J., « Long Short-Term Memory », *Neural Comput.*, vol. 9, n° 8, p. 1735-1780, November, 1997.

Hofmann T., « Unsupervised Learning by Probabilistic Latent Semantic Analysis », *Machine Learning*, vol. 42, n° 1, p. 177-196, 2001.

Howard J., Ruder S., « Universal Language Model Fine-tuning for Text Classification », *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, Association for Computational Linguistics, p. 328-339, 2018.

Kim Y., « Convolutional Neural Networks for Sentence Classification », *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, p. 1746-1751, October, 2014.

Lafferty J., McCallum A., Pereira F., « Conditional random fields : probabilistic models for segmenting and labeling sequence data », *icml*, p. 282-289, 2001.

Lavergne T., Cappé O., Yvon F., « Practical Very Large Scale CRFs », *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, Association for Computational Linguistics, Uppsala, Sweden, p. 504-513, July, 2010.

Le H.-S., Allauzen A., Yvon F., « Continuous Space Translation Models with Neural Networks », *Proceedings of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT)*, Association for Computational Linguistics, Montréal, Canada, p. 39-48, June, 2012.

Li N., Liu S., Liu Y., Zhao S., Liu M., Zhou M., « Close to Human Quality TTS with Transformer », *CoRR*, 2018.

Mikolov T., Chen K., Corrado G., Dean J., « Efficient estimation of word representations in vector space », *International Conference on Learning Representations (ICLR)*, 2013.

Nakamura M., Maruyama K., Kawabata T., Kiyohiro S., « Neural network approach to word category prediction for english texts », *Proceedings of the International Conference on Computational Linguistics (COLING)*, vol. 3, p. 213-218, 1990.

Nakamura M., Shikano K., « A study of English word category prediction based on neural networks », *The Journal of the Acoustical Society of America*, 1988.

Peters M., Neumann M., Iyyer M., Gardner M., Clark C., Lee K., Zettlemoyer L., « Deep Contextualized Word Representations », *Proceedings of the North American Chapter of*

the Association for Computational Linguistics (NAACL), Association for Computational Linguistics, p. 2227-2237, 2018.

Schuster M., Paliwal K., « Bidirectional Recurrent Neural Networks », *IEEE Transaction on Signal Processing*, vol. 45, nº 11, p. 2673-2681, November, 1997.

Schütze H., « Word Space », *Advances in Neural Information Processing Systems 5, [NIPS Conference, Denver, Colorado, USA, November 30 - December 3, 1992]*, p. 895-902, 1992.

Schwenk H., Dchelotte D., Gauvain J.-L., « Continuous space language models for statistical machine translation », *Proceedings of the International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 723-730, 2006.

Schwenk H., Gauvain J.-L., « Connectionist Language Modeling for Large Vocabulary Continuous Speech Recognition », *Proceedings of the IEEE international conference on Acoustics, speech, and signal processing (ICASSP)*, Orlando, p. 765-768, May, 2002.

Sejnowski T. J., Rosenberg C. R., NETtalk : A parallel network that learns to read aloud, Technical Report nº 86/01, Johns Hopkins University Department of Electrical Engineering and Computer Science Technical, 1986.

Sejnowski T. J., Rosenberg C. R., *Neurocomputing : Foundations of Research*, MIT Press, Cambridge, MA, USA, chapter NETtalk : A Parallel Network That Learns to Read Aloud, p. 661-672, 1988.

Teh Y. W., « A Hierarchical Bayesian Language Model based on Pitman-Yor Processes », *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, p. 985-992, 2006.

van den Oord A., Dieleman S., Zen H., Simonyan K., Vinyals O., Graves A., Kalchbrenner N., Senior A., Kavukcuoglu K., « WaveNet : A Generative Model for Raw Audio », *Arxiv*, 2016.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L. u., Polosukhin I., « Attention is All you Need », *in* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds), *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., p. 6000-6010, 2017.

Waibel A., Hanazawa T., Hinton G., Shikano K., Lang K. J., *Readings in Speech Recognition*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, chapter Phoneme Recognition Using Time-delay Neural Networks, p. 393-404, 1990.

Xu K., Ba J., Kiros R., Cho K., Courville A. C., Salakhutdinov R., Zemel R. S., Bengio Y., « Show, Attend and Tell : Neural Image Caption Generation with Visual Attention », *Proceedings of the International Conference of Machine Learning (ICML)*, 2015.

Zipf G., *The Psychobiology of Language : An Introduction to Dynamic Philology*, M.I.T. Press, Cambridge, Mass., 1935.

Classifying Semantic Clause Types With Recurrent Neural Networks: Analysis of Attention, Context & Genre Characteristics

Maria Becker* — **Michael Staniek*** — **Vivi Nastase*** — **Alexis Palmer**** — **Anette Frank***

* *Heidelberg University, Department of Computational Linguistics*
** *University of North Texas, Department of Linguistics*

ABSTRACT. *Detecting aspectual properties of clauses in the form of semantic clause types has been shown to depend on a combination of syntactic-semantic and contextual features. We explore this task in a deep-learning framework, where tuned word representations capture linguistic features. We introduce an attention mechanism that pinpoints relevant context information. Our model implicitly captures task-relevant features and avoids the need to reproduce explicit linguistic features for other languages.We present experiments for English and German that achieve competitive performance, and analyze the outputs of our systems from a linguistic point of view. We present a novel take on modeling and exploiting genre information and showcase the adaptation of our system from one language to another.*

RÉSUMÉ. *Il a été démontré que la détection des propriétés aspectuelles des clauses sous la forme de clauses sémantiques dépend d'une combinaison de caractéristiques linguistiques. Nous explorons cette tâche dans un cadre d'apprentissage sur la base des réseaux de neurones profonds. Nous introduisons un mécanisme d'attention qui identifie le contexte pertinent. Notre modèle permet d'éviter la nécessité de reproduire des caractéristiques linguistiques pour d'autres langues. Nous présentons des expériences pour l'anglais et l'allemand qui atteignent des performances compétitives, et explorons nos résultats d'un point de vue linguistique. Nous présentons une nouvelle approche pour la modélisation de l'information du genre de texte et nous mettons en valeur l'adaptation de notre système d'une langue à l'autre.*

KEYWORDS: *semantic clause types, situation entities, deep-learning, GRU, embeddings, attention mechanism, sequential information, genre, English, German.*

MOTS-CLÉS : *types de clauses sémantiques, entités de situation, deep-learning, URG, word embedding, mécanisme d'attention, information séquentielle, le genre de texte, anglais, allemand.*

TAL. Volume 59 - n° 2/2018, pages 15 à 48

1. Introduction

Semantic clause types, called *Situation Entity (SE)* types (Smith, 2003; Palmer *et al.*, 2007), are linguistic characterizations of aspectual properties shown to be useful for argumentation structure analysis (Becker *et al.*, 2016b), genre characterization (Palmer and Friedrich, 2014), and detection of generic and generalizing sentences (Friedrich and Pinkal, 2015). Recent work on automatic identification of SE-types relies on feature-based classifiers for English that have been successfully applied to various textual genres (Friedrich *et al.*, 2016). Sophisticated features have been built by the prior work to capture diverse linguistic indicators of SE-types, including morpho-syntactic and rich semantic features. The larger context was also shown to be useful: Friedrich *et al.* (2016) used a sequence labeling approach that took into account contextual clause labels, leading to improved classification performance.

Deep learning provides a powerful framework in which linguistic and semantic regularities can be implicitly captured (to a certain degree) through word embeddings (Mikolov *et al.*, 2013b). Also, neural systems are able to detect features useful for a given task while learning takes place, through error back-propagation (Goodfellow *et al.*, 2016; Goldberg, 2017). Patterns in larger text fragments can be encoded and exploited by recurrent (RNNs) or convolutional neural networks (CNNs) which have been successfully used for various sentence-based classification tasks, e.g., sentiment (Kim, 2014) or relation classification (Vu *et al.*, 2016; Tai *et al.*, 2015).

We frame the task of classifying clauses with respect to their aspectual properties – i.e., SE-types – in a recurrent neural network architecture. We adopt a Gated Recurrent Unit (GRU)-based RNN architecture that is well suited to modeling long sequences (Yin *et al.*, 2017). This initial model is enhanced with an attention mechanism shown to be beneficial for sentence classification (Wang *et al.*, 2016) and sequence modeling (Dong and Lapata, 2016). We explore the usefulness of attention in two settings: (i) the individual classification task, and (ii) a setting approximating sequential labeling in which the attention vector provides features that describe the clauses preceding the target instance. Compared to the strong baseline provided by the feature-based system of Friedrich *et al.* (2016), we achieve competitive performance and find that attention, context representation using labels of previous clauses, and information about the text genre significantly improve our model.

A strong motivation for developing NN-based systems is that they can be transferred with low cost to other languages without major feature engineering or use of hand-crafted linguistic knowledge resources. Given the highly engineered feature sets used for SE classification so far (Friedrich *et al.*, 2016), porting such classifiers to other languages is a non-trivial issue. We test the portability of our system by applying it to German. Since our system is supervised, this presupposes that annotated training data is available.

A downside of neural models is of course that they are relatively opaque with respect to the nature of the learned features. We try to counter this weakness by deeper model analysis: by exploiting the weights of the learned attention vectors and by

relating genre-specific linguistic properties to the classification performance of genre-aware SE classification models.

Our work presents a novel take on modeling and exploiting genre information for the task of SE classification. We test our models on the English multi-genre corpus of Friedrich *et al.* (2016) and on a German multi-genre corpus which we assembled for that purpose. We provide qualitative evaluation and investigation of the learned models, by relating learned attention weights of the model to different linguistic attributes: POS classes, individual word tokens and positional information. We also investigate genre-specific information, such as frequent SE-type n-grams in different genres, and relate them to the performance of the genre-aware SE classification models.

Our aims and contributions are: (i) We study the performance of GRU-based models enhanced with attention over various window sizes for modeling local and non-local characteristics of semantic clause types; (ii) we compare the effectiveness of the learned attention weights as features for a sequence labeling system to the explicitly defined syntactic-semantic features in Friedrich *et al.* (2016); (iii) we define model extensions that integrate external knowledge about genre and show that this improves classification performance across genres; (iv) we test the portability of our models to other languages by applying them to a smaller, manually annotated German dataset and show that the performance is comparable to English. (v) We perform qualitative evaluation based on learned attention weights and distributional information on genre.

In what follows, Section 2 introduces the linguistic categories of semantic clause types as used in our work. Section 3 situates our contribution in relation to prior work on linguistic and computational aspects of SE-type classification. Section 4 proposes GRU-based model variants for SE-type classification, including local and context-informed models, models that incorporate attention over token embeddings or predicted SE-type labels in the previous context, and models that make use of external genre information as an additional feature. In Section 5 and 6, we describe our experimental data and settings, how we train and evaluate our models, and report and compare results. Section 7 presents a deeper investigation of the learned attention weights and the impact of textual genre for SE classification. In Section 8 we transfer our system to annotated data for German and analyze its performance, also in relation to the English classifier. Section 9 summarizes and concludes with perspectives on future work.

2. Semantic Clause Types

Situation entities were identified by Smith (2003) as one of the linguistic correlates to variations in text type at the level of the text passage. In other words, modes of discourse such as *narrative* and *argument/commentary* are distinguishable from one another by readers in part because of their varying distributions of situation types (or semantic clause types). Narrative passages consist primarily of events and states, for example, and argumentative passages make heavy use of generics and generalizing

sentences. These observations have since been supported by further empirical investigations (Becker *et al.*, 2016a; Mavridou *et al.*, 2015; Palmer and Friedrich, 2014).

Semantic clause types can be distinguished by the function they have within a text or discourse. We use the inventory of semantic clause types, also known as **situation entity (SE) types**, developed by Smith (2003) and extended in Palmer *et al.* (2007) and Friedrich and Palmer (2014b). SE-types describe the abstract semantic types of situations evoked in discourse through clauses of text. As such, they capture the manner of presentation of the content, along with the information content itself. For example, some propositional content can be alternately described with a focus on the eventive aspect of the proposition (*The car squealed around the corner*) or on the stative aspects of the proposition (*The car's squeal was deafeningly loud*).

The seven SE-types we use are described below. The first subset – eventualities – consists of **states**, **events**, and **reports**. Report-type entities (e.g., the italicized portion of (3) below) typically provide attribution for statements and are modeled as a subtype of event.

1) STATE (S): *Armin has brown eyes.*
2) EVENT (EV): *Bonnie ate three tacos.*
3) REPORT (R) provides attribution: *The agency said* costs had increased.

Further SE categories are **generic sentences** and **generalizing sentences** (sometimes referred to as habituals). The former predicate over classes or kinds; the latter describe regularly-occurring events, such as habits of individuals.

4) GENERIC SENTENCE (GEN): *Birds can fly. – Scientists make arguments.*
5) GENERALIZING SENTENCE (GS): *Fei travels to India every year.*

The final two SE-types included in our inventory are QUESTION and IMPERATIVE.

6) QUESTION (Q): *Why do you torment me so?*
7) IMPERATIVE (IMP): *Listen to this!*

An eighth class OTHER is assigned to clauses without a SE-label, e.g., bylines or email headers.

Semantic features for SE-type. Determining the SE-type is a complex task involving interactions between lexical and grammatical information, syntactic structure, and various aspectual and other semantic features. In particular, three classes of semantic feature have been identified as useful for identifying the SE-type of a clause (Friedrich and Palmer, 2014b): the *lexical aspectual class (stative or dynamic) of the clause's main verb*, *habituality of the clause*, and the *nature of the main referent of the clause*. [1] An especially useful main referent feature is the *genericity of the main referent* – whether or not it evokes a class or kind.

1. The main referent of a clause is roughly the person/thing/situation the clause is about, often realized as its grammatical subject.

Motivation. The semantic distinctions made by this inventory of SE-types are linguistic in nature, and each individual SE category has been well-studied in linguistic theory; please see Smith (2005) for an extensive list of relevant literature. The particular inventory is motivated by theoretical work which aims to understand the nature of text type. Smith (2003) assembles this set of clause types following investigation of the linguistic differences between text passages of different Discourse Modes (e.g., argumentative, narrative, reporting), because these clause types, together with mode of progression, explain distinctions between text types. In addition, they allow for near-exhaustive annotation of the clauses of an individual text passage. In this work, we use the inventory described on the previous page, which is a subset of Smith's inventory, leaving out only two infrequently-occurring types in the category of ABSTRACT ENTITIES (Friedrich *et al.*, 2016).

The ability to classify clauses by SE +-type lays the foundation for automatic classification of text passages according to Discourse Mode (see, for example, Song *et al.* (2017)). In addition to their role in text type classification, SE-types have been shown to be useful for determination of event duration (Vempala *et al.*, 2018; Sanagavarapu *et al.*, 2017). Additional applications are anticipated in temporal interpretation, event extraction, and narrative analysis as well as for the extraction of knowledge, e.g., generalizing knowledge (Reiter and Frank, 2010). Moreover, SE-types play a role in argumentation structure analysis (Becker *et al.*, 2016b) and have been shown to be useful for genre characterization (Palmer and Friedrich, 2014).

3. Related Work

Semantic clause types and text passages. The use of linguistic features for distinguishing text passages is closely related to Argumentative Zoning (Teufel, 2000; O'Seaghdha and Teufel, 2014), where linguistic features are used to distinguish genre-specific types of text passages in scientific texts. In this manner, those texts are segmented into types of text passages such as Methods or Results. There is a correlation between the distribution of SE-types in text passages and discourse modes, e.g., narrative, informative, or argumentative (Palmer and Friedrich, 2014; Mavridou *et al.*, 2015; Becker *et al.*, 2016a). Notions related to SE-types have been widely studied in theoretical linguistics (Vendler, 1957; Verkuyl, 1972; Dowty, 1979; Smith, 1991; Asher, 1993; Carlson and Pelletier, 1995) and have seen growing interest in computational linguistics (Siegel and McKeown, 2000; Zarcone and Lenci, 2008; Herbelot and Copestake, 2009; Reiter and Frank, 2010; Costa and Branco, 2012; Nedoluzhko, 2013; Friedrich and Palmer, 2014a; Friedrich and Pinkal, 2015; Song *et al.*, 2017).

Feature-based classification of SE-types. The first robust system for SE-type classification (Friedrich *et al.*, 2016) combines task-specific syntactic and semantic features with distributional word features, as captured by Brown clusters (Brown *et al.*, 1992). Syntactic features include (among others) selected structural configurations associated with SE-type, as well as dependency relations associated with the main referent. Semantic features include (among others) WordNet senses, countability, and

presence of negation and/or modality. This system segments each text into a sequence of clauses and then predicts the best sequence of SE-labels for the text using a linear chain conditional random field (CRF) with label bigram features. [2]

Although SE-types are relevant across languages, their linguistic realization varies across languages. Accordingly, some of Friedrich *et al.* (2016)'s syntactic and semantic features are language-specific and are extracted using English-specific resources such as WordNet and Loaiciga *et al.* (2014)'s rules for extracting tense and voice information from POS tag sequences.

Friedrich *et al.* (2016)'s system is trained and evaluated on data sets from MASC and Wikipedia (cf. Section 5), reaching accuracies of 76.4% (F1 71.2) with 10-fold cross-validation, and 74.7% (F1 69.3) on a held-out test set. To evaluate the contribution of sequence information, Friedrich *et al.* (2016) compare the CRF model to a Maximum Entropy baseline, noting that the sequential model significantly outperforms the model which classifies clauses in isolation, particularly for the less-frequent SE-types of GENERIC SENTENCE and GENERALIZING SENTENCE.

When trained and tested within a single genre (of the 13 genres represented in the data sets), Friedrich *et al.* (2016)'s system performance ranges from 26.6 F1 (for government documents) to 66.2 F1 (for jokes). Training on all genres levels out this performance difference, with a range of F1 scores from 58.1 to 69.8. This shows that their classifiers generalize over the different genres present in the dataset. However, genre information is not explicitly modeled in their approach.

Neural approaches to sentence classification, sequence and context modeling. Inspired by research in vision, sentence classification tasks have initially been modeled using Convolutional Neural Networks (Kim, 2014; Kalchbrenner *et al.*, 2014; Mishra *et al.*, 2017) which are particularly suitable for tasks that rely on discovering patterns that are distributed over the input signal. RNN variations – with Gated Recurrent Units (GRU) (Cho *et al.*, 2014; Abdul-Mageed and Ungar, 2017) or Long Short-Term Memory units (LSTM) (Hochreiter and Schmidhuber, 1997) – have since achieved state-of-the-art performance in both sequence modeling and classification tasks. Recent work applies bi-LSTM models in sequence modeling (PoS tagging (Plank *et al.*, 2016), NER (Lample *et al.*, 2016)) and structure prediction tasks (Semantic Role Labeling (Zhou and Xu, 2015) or semantic parsing into logical forms (Dong and Lapata, 2016)). Sentence representation learning from specifically selected training data has also been done using bi-LSTM models (Conneau *et al.*, 2017; Nie *et al.*, 2017). Tree-based LSTM models have been shown to often perform better than purely sequential bi-LSTMs (Tai *et al.*, 2015; Miwa and Bansal, 2016; Cheng and Miyao, 2017), but depend on parsed input.

Hierarchical classification models. Song *et al.* (2017) develop a neural hierarchical multi-class sequence labeling model for automatic labeling of Discourse Modes in essays. Their system uses a sentence-level GRU layer for sentence encoding and a bi-

2. Code and data: https://github.com/annefried/sitent.

GRU layer to connect the encoded sentences and to perform sequence prediction for discourse mode labeling. They use this model to improve automatic essay scoring in Chinese using the predicted discourse mode labels as features. Their work is related to, but by-passes, the level of SE classification. Their model does not make use of the attention mechanism and does not offer a deeper linguistic analysis of the learned models.
Song *et al.* (2017) observe that accessing information about past and future sentences provides more contextual information for current discourse mode prediction, which is in line in with our hypothesis that modeling contextual information yields improved performance for classifying semantic clause types. The model we propose in Section 4 incorporates context information by using separate GRUs and predicts the SE-type for one clause each time. Inspired by Song *et al.* (2017)'s work, we leave a model which jointly learns representations for sequences of clauses in a text or a paragraph as future work. [3]

Attention. Attention has been established as an effective mechanism that allows models to focus on specific words in the larger context. A model with attention learns what input tokens or token sequences to attend to and thus does not need to capture the complete input information in its hidden state. Attention has been used successfully e.g., in aspect-based sentiment classification (Wang *et al.*, 2016), for modeling relations between words or phrases in encoder-decoder models for translation (Bahdanau *et al.*, 2015), or bi-clausal classification tasks such as textual entailment (Rocktäschel *et al.*, 2016). We make use of attention to larger context windows and previous labeling decisions to capture sequential information relevant for our classification task, and we investigate the learned weights to gain insights about what the models learn.

4. Models

We aim for a system that can fine-tune input word embeddings to the task, and that can process clauses as sequences of words from which to encode larger patterns that help our particular clause classification task. GRU RNNs are used because they can process successfully long sequences and capture long-term dependencies. Attention can encode which parts of the input contain relevant information. These modeling choices are described and justified in detail below.

3. During the revision phase of this article, Dai and Huang (2018) published a hierarchical neural model for SE classification. They design a unified neural network which models word-level dependencies and clause-level dependencies jointly in order to derive clause representations for SE-type prediction. When being trained on the English dataset which we also use in our work, this model achieves up to 80.7 accuracy on the test set, beating all of the baseline models. We expect that adopting a hierarchical classification framework will result in further improvement of our results.

4.1. *Model Components*

4.1.1. *Basic Model: Gated Recurrent Unit*

Recurrent Neural Networks (RNNs) are modifications of feed-forward neural networks with recurrent connections, which allow them to find patterns in – and thus model – sequences. The latter makes the representations suitable for our task, given that we aim to capture sequence information. Simple RNNs cannot capture long-term dependencies (Bengio *et al.*, 1994) because the gradients tend to vanish or grow out of control with long sequences. Gated Recurrent Unit (GRU) RNNs, proposed by Cho *et al.* (2014), address this shortcoming. GRUs have fewer parameters and thus need less data to generalize (Zhou *et al.*, 2016) compared to LSTM RNNs, and also outperform the LSTM in many cases (Yin *et al.*, 2017), which makes them a good choice for our relatively small dataset. Comparison of GRUs, bi-GRUs, LSTMs and bi-LSTMs on our dataset for our classification task – in initial experiments, not reported here – showed that GRUs outperform the other three, confirming this hypothesis.

The relevant equations for a GRU are given below. x_t is the input at time t (usually a dense word embedding vector), r_t is a reset gate which determines how to combine the new input with the previous memory, and the update gate z_t defines how much of the previous memory to keep. h_t is the hidden state (memory) at time t, and $\tilde{h_t}$ is the candidate activation at time t. W_* and U_* are weights that are learned. $\odot$ denotes the element-wise multiplication of two vectors.

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad [1]$$

$$\tilde{h_t} = tanh(W x_t + U(r_t \odot h_{t-1})) \quad [2]$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad [3]$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h_t} \quad [4]$$

The last hidden vector h_T (with T the number of tokens in the input clause) will be taken as the representation of the input clause. After compressing it into a vector whose length is equal to the number of class labels (=8) using a fully connected layer with sigmoid function, we apply $softmax$ to transform it to a probability distribution.

4.1.2. *Neural Attention Mechanism*

We extend our GRU model with a neural attention mechanism to capture the most relevant words in the input clauses for classifying SE-types. Specifically, we adapt the base implementation of attention used in Rocktäschel *et al.* (2016) for our clause classification task as follows:

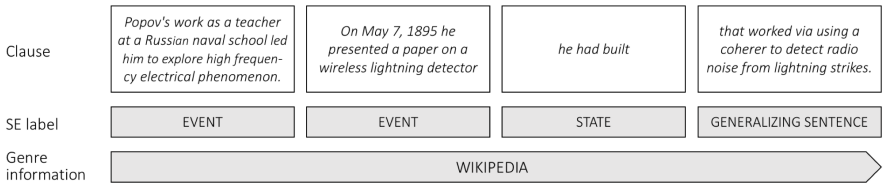


Figure 1. *An example from Wikipedia illustrating context and genre information modeled in our system. Assuming "he had built" is the clause to be classified (with label* STATE*), context and genre information can be taken into account in different ways: the* ***token context model*** **CON_TOK2+GEN** *uses all tokens of the previous two clauses jointly with information about the genre (here, Wikipedia); the* ***label context model*** **CON_LAB2+GEN** *instead uses as input the target clause and the (predicted) labels of the two previous clauses (*EVENT, EVENT*) jointly with genre information.* ***Local models*** *would use only the target clause token inputs (optionally jointly with genre information) for classification.*

$$M = tanh(W_h H + W_v h_T \otimes e_T) \quad [5]$$

$$\alpha = softmax(w^T M) \quad [6]$$

$$r = H\alpha^T \quad [7]$$

where H is a matrix consisting of the hidden vectors $[h_1, ..., h_T]$ produced by the GRU, h_T is the last output vector of the GRU, and e_T is a vector of 1s where T denotes the T tokens in the input clause. $\otimes$ denotes the outer product of the two vectors. α is a vector consisting of attention weights and r is a weighted representation of the input clause. W_h, W_v, and w are parameters to be learned during training.

The final clause representation is obtained from a combination of the attention-weighted representation r of the clause and the last output vector h_T.

$$h^* = tanh(W_p r + W_x h_T) \quad [8]$$

where W_p and W_x are trained projection matrices. We convert h^* to a real-valued vector of length 8 (the number of target classes) and apply $softmax$ in order to transform it to a probability distribution over the 8 output classes, i.e., the predicted SE-types.

4.1.3. *Modeling Context and Genre Information*

Previous analyses show that text types differ with respect to their SE-type distributions (Friedrich and Pinkal, 2015). Furthermore, specific n-grams over SE-types are

more frequent within some textual genres than in others. This supports the choice of incorporating (sequential) context information and information about genre as additional features for the classification of SE-types. The English corpus we use consists of texts from 13 genres; the German corpus covers 7 genres. Figure 1 illustrates both the context and the genre information that our models consider for classifying SE-types.

4.2. *Model Types*

We investigate different model types: Local Models and Context Models.

4.2.1. *Local Models*

LOC, LOC_ATT and LOC_ATT+GEN. We first experiment with models that only consider the local clause for SE classification. Adding attention mechanism and genre information to our basic local model results in three versions of the local model: (i) **basic local model** (LOC) uses as input only the hidden representation computed over the tokens of the clause to be classified; the last hidden vector $[h_T]$ (h_T from Equation 4) is passed through the fully connected layer and softmax; (ii) **local model enhanced with attention** to the representations of the tokens in the local clause (LOC_ATT); here $[h^*]$ (h^* as defined in Equation 8) is used for projection; and (iii) **local model enhanced with attention and genre information** (LOC_ATT+GEN), where genre information is encoded as a dense embedding g of a genre label which is initialized randomly [4]; here we concatenate the attention vector h^* and a genre label embedding g $[h^*; g]$. Illustrations for all three model types are given in Figure 2.

4.2.2. *Context Models*

We also investigate models that consider not only the local clause for SE classification in model training, but also the previous clauses' token sequences or their labels. We experiment with several settings:

1) different window sizes of token sequences or of previous labels

2) applying or omitting attention to token sequences or previous labels

3) adding or omitting genre information.

These settings result in various model combinations. In the interest of clarity and space, we only describe and report the results for our best performing models for the following three categories: (i) context models using *tokens* of previous clauses jointly with genre information; (ii) context models using *labels* of previous clauses jointly with genre information; and (iii) context models using *tokens and labels* of previous clauses jointly with genre information.

4. We also use a GRU for the genre representation. This was a design decision in order to keep representations uniform. Note that the individual GRUs (tokens, labels, genre) do not share parameters.

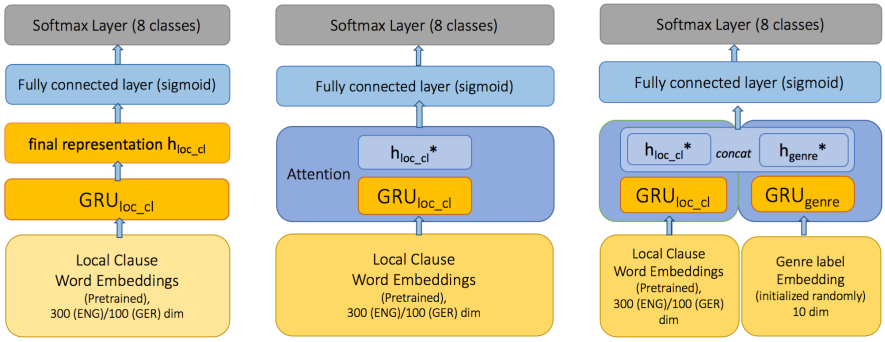


Figure 2. *Architecture of our local models: basic local model without attention (*LOC*, left), local model with attention (*LOC_ATT*, middle) and local model with attention using genre information (*LOC_ATT+GEN*, right).*

CON_TOK+GEN. When considering **tokens of previous clauses**, we add one GRU model for each previous clause ($h_1; h_2; ...; h_N$, with N the number of previous clauses) and concatenate their final outputs with the final output of the GRU with attention for the target clause h_0^* and with the final output of the GRU for genre label encoding h_g (cf. Figure 3). [5]

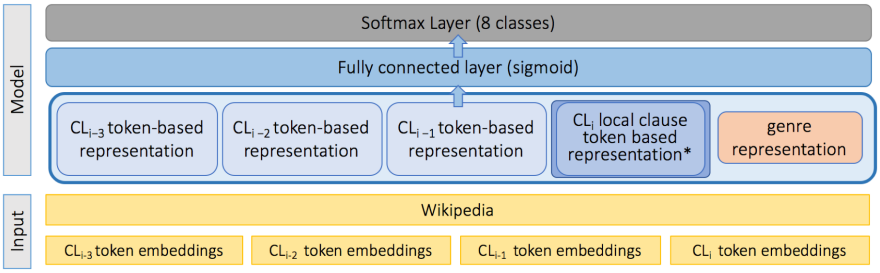


Figure 3. *Context model using tokens of target clause (with attention), tokens of previous clauses (without attention) with N=3, jointly with genre information (*CON_TOK+GEN*).*

In our experiments we found that models perform best when we apply the attention mechanism only to the GRU for the target clause itself (instead of applying it also to the GRUs for the previous clauses).

$$h^*_{con\_tok+gen} = [h_1; h_2; ...; h_N; h_0^*; h_g] \quad [9]$$

5. The concatenation operation is denoted by square brackets, and the elements which are concatenated are separated by semicolons.

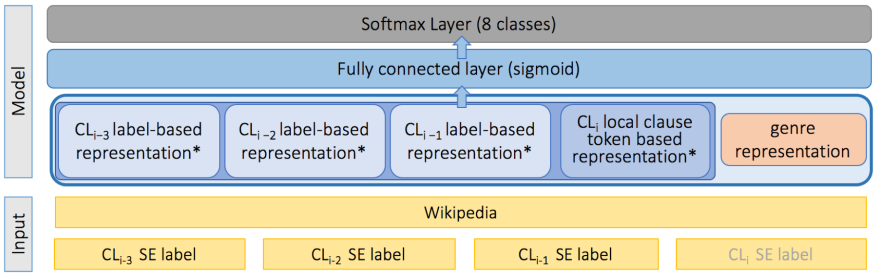


Figure 4. *Context model using tokens of target clause (with attention), labels of previous clauses (with attention) with N=3, jointly with genre information (*CON_LAB+GEN*).*

We then transform the concatenated vector into a dense vector equal to the number of class labels and apply $softmax$ to predict the local clause's SE-label.

CON_LAB+GEN. For including **labels of the previous clauses** in our model, we first transform the gold-standard labels used during training into embeddings, concatenate them and apply attention to the sequence of labels (i.e., to the concatenation of the label vectors). We then concatenate the final hidden state of the target clause with the attention vector learned over the sequence of labels of the previous clauses and with the final output of the GRU for genre, cf. Figure 4:

$$h^*_{con\_lab+gen} = [h^*_{lab}; h^*_0; h_g] \quad [10]$$

where h^*_{lab} is the last hidden state from the GRU used on the label sequence of previous labels over which attention is applied jointly ($h^*_{lab} = [h_1; h_2; ...; h_N]^*$), h^*_0 is the final output of the GRU with attention for the target clause, and h_g is the final output of the GRU for genre. At test time, we use the predicted probability distribution vector of the labels of the previous clauses.

$$h^*_{con\_toklab+gen} = [h_1; h_2; ...; h_N; h_{lab}; h_0; h_g] \quad [11]$$

CON_TOKLAB+GEN. We also perform experiments that include both the embedding representations for tokens and for the labels of previous clauses. One GRU model is added for each of the previous clauses (tokens), their final outputs $h_1; h_2; ...; h_N$ are then concatenated with the embeddings for the labels of the previous clauses h_{lab}, with the final output of the GRU for the target clause h_0, and with the final output of the GRU for genre h_g. This model performs best when the attention mechanism is omitted. It is illustrated in Figure 5.

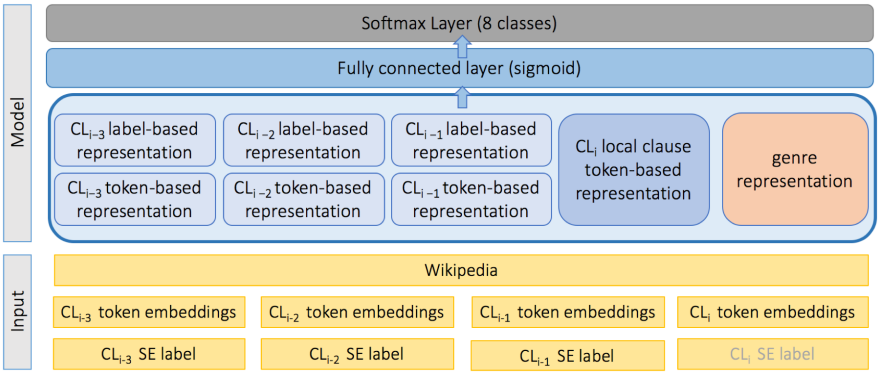


Figure 5. *Context Model using tokens of target clause (without attention), tokens and labels of previous clauses (both without attention) with N=3, jointly with genre information (*CON_TOKLAB+GEN*).*

5. Data

5.1. *Datasets*

We use the English dataset described in Friedrich and Palmer (2014b).[6] The texts, obtained from Wikipedia and MASC (Ide *et al.*, 2010), range across 13 genres, e.g., news texts, government documents, essays, fiction, jokes, emails. For German, we combine three data sets described in Mavridou *et al.* (2015), Becker *et al.* (2016a) and Becker *et al.* (2017).[7] The German texts cover 7 genres: argumentative essays (Peldszus and Stede, 2015), Wikipedia articles, fiction, commentary, news texts, TED talks, and economic reports. Statistics are given in Table 1.

Data set	# Instances (Clauses)	# Tokens
English: MASC	30,333	357,078
English: Wiki	10,607	148,040
German: all	18,194	236,522

Table 1. *Datasets with SE-labeled clauses*

Figure 6 gives an overview of the distribution of instances (i.e., clauses) among genres within our English and German datasets. Compared to the English dataset, the German dataset is smaller (44% in size) and less diverse with respect to genre

6. Available at: https://github.com/annefried/sitent.

7. Available at: http://www.cl.uni-heidelberg.de/english/research/downloads/resource_pages/GER_SET/GER_SET_data.shtml.

(7 instead of 13 genres). The genres in the German dataset are more similar to one another than those in the English dataset.

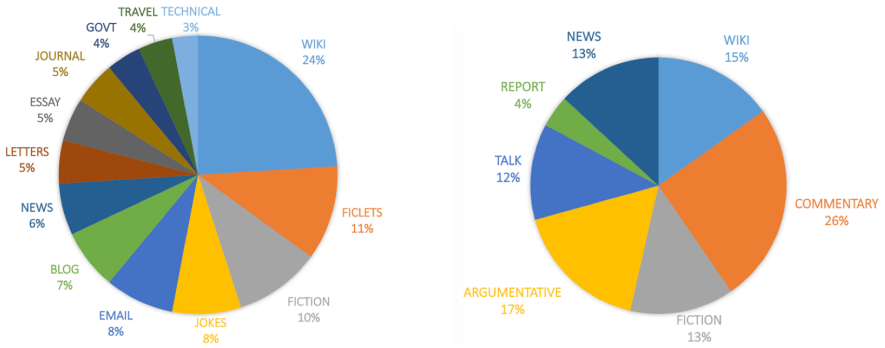


Figure 6. *Distribution of instances among genres within our English (left) and German (right) datasets.*

5.2. *Distribution of SE Types and their N-grams among Different Genres*

Text types differ in their SE-type distributions: Palmer and Friedrich (2014) find that GENERIC SENTENCES and GENERALIZING SENTENCES play a predominant role for texts associated with the argument or commentary mode (such as essays), and EVENTS and STATES for texts associated with the report mode (such as news texts). Becker *et al.* (2016a) find that argumentative texts are characterized by a high proportion of GENERIC and GENERALIZING SENTENCES and very few EVENTS, while reports and talks contain a high proportion of STATES, and fiction is characterized by a high number of EVENTS.

The distribution of SE-types in our datasets. When analyzing our data, we observe a striking difference between Wikipedia articles and other genres regarding the distribution of SE-types (cf. Figure 7). For the selected English Wikipedia texts, 50% of the SE-types are GENERIC SENTENCE clauses, with STATES second at 24.3%.[8] For the 12 MASC genres, STATE is the most frequent type (49.8%), with EVENTS second at 24.3%. GENERIC SENTENCES make up only 7.3% of the SE-types in the MASC texts. In the German data, the distribution of SE-types also differs according to genre: in argumentative texts, for example, GENERIC SENTENCES make up 48% of the SE-types, followed by STATES with a proportion of 32%, while in most other genres the most frequent class is STATE.

8. The Wikipedia texts were selected by Friedrich *et al.* (2015) precisely in order to target GENERIC SENTENCE clauses.

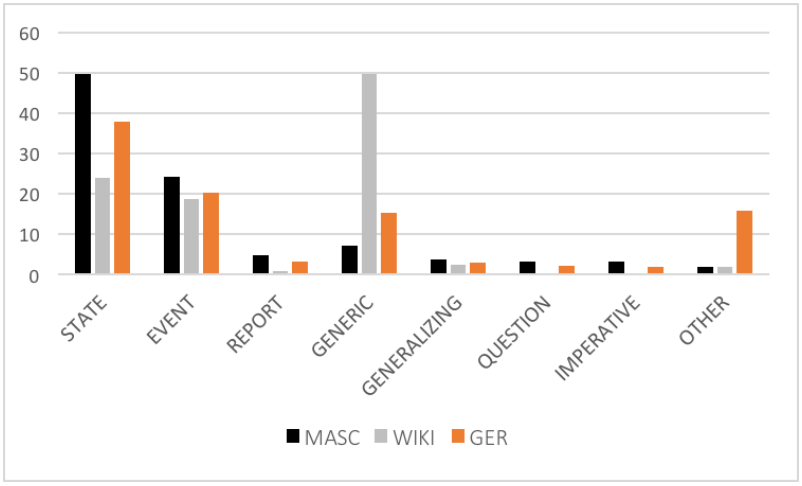


Figure 7. *Distribution of SE-types within different textual genres included in the corpus (in percentage). MASC=English corpus consisting of 12 genres; WIKI=English corpus consisting of Wikipedia articles, GER=German corpus consisting of 7 different genres.*

We also find that *sequences of SE-types* differ among genres: e.g., while STATE-STATE is the most frequent bigram within journal articles, the most frequent bigram in Wikipedia articles is GEN-GEN.[9] The most frequent trigram in jokes is EVENT-EVENT-EVENT, followed by STATE-STATE-STATE, whereas in government documents the most frequent trigrams are STATE-STATE-STATE and EVENT-STATE-STATE. In Section 7, we will further analyze such differences in the distribution of SE-label sequences across genres and how these differences are reflected by the performance of different model types.

5.3. *Confusability of Classes*

Some SE-types capture subtle aspectual distinctions, and as such are easily confused with one another. Friedrich (2017) (p. 93) provides a detailed analysis of annotator coincidence for the two English datasets used here. The three most inconsistently labeled types are STATES, GENERIC SENTENCES, and GENERALIZING SENTENCES. GENERALIZING SENTENCES are often labeled as EVENT or GENERIC SENTENCE; STATES are often labeled as GENERIC SENTENCE; and the label STATE is often applied to clauses labeled by other annotators as GENERIC SENTENCE, GENERALIZING SENTENCE, or EVENT.

9. GEN abbreviates GENERIC SENTENCE.

	Dataset	Eval	Acc	F1
Palmer07	Brown data	test	53.1	-
Fried16, set A (CRF)	MASC+Wiki	test	69.8	63.9
Fried16, set B (CRF)	MASC+Wiki	test	71.4	65.5
Fried16, set A+B (CRF)	MASC+Wiki	test	74.7	69.3
Fried16, set A+B (CRF)	MASC+Wiki	CV	76.4	71.2
Fried16, set A+B (CRF, seq-oracle)	MASC+Wiki	CV	77.9	73.9
BoW + SVM	MASC+Wiki	test	64.8	47.3

Table 2. *Reported results of baseline models for English from Palmer et al. 2007, Friedrich et al. 2016 and our own baseline system using word unigrams and bigrams as input for a SVM classifier (accuracy and macro-average F1 score). CV=10-fold cross-validation, test = evaluation on test set (20% of dataset, distinct set of documents in train and test). Since CV splits are not available, we only compare to the results on the held-out test set.*

5.4. *Segmentation*

The texts of the English dataset have been split into clauses using SPADE (Soricut and Marcu, 2003) with some heuristic post-processing. For the German dataset, DiscourseSegmenter's rule-based segmenter (EDSEG, Sidarenka *et al.* (2015)) was used. It uses German-specific rules to determine the boundaries of elementary discourse units in texts. Because DiscourseSegmenter occasionally oversplits segments, a small amount of post-processing was performed.

6. Experiments and Evaluation

6.1. *Baseline Systems*

From earlier work, the feature-based system of Palmer *et al.* (2007) (*Palmer07* in Table 2) simulates context through predicted labels from previous clauses. Their results are reported on 20 texts from the *popular lore* section of the Brown corpus (Francis and Kucera, 1979). Friedrich *et al.* (2016) (*Fried16* in Table 2) report results for their CRF-based SE-type labeler for different feature sets, evaluating both with 10-fold cross-validation and on a held-out test set (20% of the dataset, with distinct sets of documents in both train and test data). Training and testing were done on the combined MASC+Wiki dataset. *Fried16* is a sequence model which aims to learn the optimal global sequence of labels, jointly predicting labels for all clauses in a document. In the *oracle* setting, it includes the gold label of the previous clause. In our experiments, we adopt the models of *Fried16* as a very strong baseline for benchmarking our models, given that we are working on the same data.

Fried16's feature set A consists of standard NLP features including POS tags and Brown clusters. Feature set B includes more detailed features such as tense, lemma, negation, modality, WordNet sense, WordNet supersense and WordNet hypernym sense. We presume that some of the information captured by feature set B, particularly sense and hypernym information, as well as syntactic features, may not be captured in the word embeddings we use in our approach.

We also implement a simple baseline system which uses the CountVectorizer class from Sklearn (Pedregosa *et al.*, 2011) to calculate a bag of words matrix for the whole training and test data. Word unigrams and bigrams are used, and the resulting matrices are then fed into a LinearSVC classifier with default parameters.

Table 2 shows that, while *Palmer07* achieve modest results on Brown data, our BoW+SVM baseline is clearly lower than either of the feature-based CRF models of *Fried16* (on test set or 10-fold cross-validation setup). *Fried16's* results show that, when using sets A and B individually, Set B performs better than Set A on held-out test set, while their combination increases performance up to 74.7 accuracy. The result for the cross-validation setup (using feature set A+B) is very close to the seq-oracle result which includes the gold label of the previous clause. *Fried16* don't report seq-oracle results for the held-out test set. Please note that a direct comparison of our results to the results of the cross-validation setup of *Fried16* is not possible.

6.2. *Model Implementation and Training Setup*

Model implementation. We implemented the model variants for our SE-type classifier as described in Section 4, using Theano (Theano Development Team, 2016). We train the model with categorical cross entropy as loss function. For feature encoding (both SE labels in the context window and the genre label), we used 10-dimensional embedding vectors that we initialized randomly.

Test-train split. For the English dataset, we use the same test-train split as Friedrich *et al.* (2016). [10] The German dataset was split into training and testing with a balanced distribution of genres (as is the case for the English dataset). Both datasets have a 80-20 split between training and testing, with 20% of training used for development (cf. Table 1). We report results in terms of accuracy and macro-average F1 score on the held-out test set.

Parameters and tuning. Hyperparameter settings were determined through exhaustive random search using *optunity* (Bergstra and Bengio, 2012) on the development set, and we use the best setting for evaluating on the test set. We tune batch size, number of layers, GRU cell size, and regularization parameter (L2). For learning rate optimization, we use AdaGrad (Duchi *et al.*, 2011) and tune the initial learning rate. For LOC, the best result on the development set is achieved for GRU with batch size 100, 2 layers, cell size 350, learning rate 0.05, and L2 regularization parameter (0.01).

10. The cross-validation splits of the data used by Friedrich *et al.* (2016) are not available.

For LOC_ATT the parameters are identical except for L2 (0.0001). We then apply the same hyperparameters as for LOC_ATT to the local model LOC_ATT+GEN and to the context models CON_TOK+GEN, CON_LAB+GEN and CON_LABTOK+GEN.

Window size as hyper-parameter. In the setting which includes previous labels we observe that the larger the window, the higher the accuracy. The opposite is the case for the context model which includes the *tokens* of previous clauses. We achieve best results when incorporating *five* previous clause labels in the CON_LAB* models or the tokens of *a single* previous clause in the CON_TOK* model (cf. Table 3). This also holds when porting our system to German (cf. Table 5). Adding more than five previous labels (or clauses) doesn't improve the system further.

Word embeddings. Word embeddings have been shown to capture syntactic and semantic regularities (Mikolov *et al.*, 2013b) and to benefit from fine tuning for specific tasks. The features used by Friedrich *et al.* (2016) cover a variety of syntactic and semantic features – such as tense, voice, number, POS, semantic clusters –, some of which we expect to be encoded in pre-trained embeddings, while others will emerge through model training. We start with pre-trained embeddings for both English and German, because this leads to better results than random initialization which we trace back to the fact that our training data isn't large enough to derive good word embeddings. For German, we use 100-dimensional word2vec embeddings trained on a large German corpus of 116 million sentences using Skip-Gram mode with 5 negative samples (Reimers *et al.*, 2014). [11] For English, we use 300-dimensional word2vec embeddings (Mikolov *et al.*, 2013a) trained on a portion of the Google News dataset (about 100 billion words). The pre-trained embeddings are tuned during training. [12]

Testing for significance. To test significance of differences in accuracy, we apply McNemar's test with $p < 0.05$ and $p < 0.01$ rejecting the null hypothesis. Here we report significant differences between the best models (based on accuracy) for each of the four categories: local models, context models using tokens of previous clauses, context models using labels of previous clauses, and context models using both tokens and labels of previous clauses. When reporting the results, a pair of models that are significantly different from each other will be marked with the same symbol respectively for $p < 0.05$ and $p < 0.01$.

6.3. *Results*

Evaluation. We present the performance of the different models proposed in Section 4 in Table 3, reporting accuracy and macro F1 score on the test set. LOC achieves an accuracy of 66.55. Adding *attention* (LOC_ATT) yields an improvement of 2.63

11. https://public.ukp.informatik.tu-darmstadt.de /reimers/2014_german_embeddings.

12. We also experimented with FastText embeddings (Joulin *et al.*, 2017). Those embeddings take into account the internal structure of words which is especially useful for morphologically rich languages. We ran the local models (LOC, LOC_ATT and LOC_ATT+GEN) with FastText embeddings and found that word2vec embeddings work slightly better.

percentage points (pp). Using both *attention and genre* information (LOC_ATT+GEN) leads to a 1.94 pp increase over the model that uses only attention (LOC_ATT). Adding **context information** beyond the local clause in the form of the embedding representations of the **tokens of previous clauses** (CON_TOK+GEN) improves the model slightly, and a smaller window size yields better results than a larger one. The best results of this model type are obtained with the model which uses only the tokens of one previous clause jointly with genre information (CON_TOK1+GEN), and where the attention mechanism is applied only to the GRU of the target clause to be classified (71.67% accuracy). Using context in the form of **predicted labels of previous clauses** (CON_LAB+GEN) also improves the model. The model which uses the predicted labels of five previous clauses together with genre information (CON_LAB5+GEN) – with the attention mechanism being applied to the GRU of the target clause to be classified and to the representation of the previous labels – is our best performing model in general and yields an accuracy of 72.04.

The results in Table 3 show that using context information in the form of **predicted labels of previous clauses and embeddings for the tokens of previous clauses** in the CON_TOKLAB+GEN model is not favorable: accuracy drops compared to CON_TOK+GEN and CON_LAB+GEN, which use these two sources of contextual information separately.

Comparison to the CRF baseline model. All of our models outperform our simple baseline system BOW BL which uses word unigrams and bigrams as input for a SVM classifier. Both the CON_TOK1+GEN model and the CON_LAB5+GEN model outperform Friedrich *et al.* (2016)'s results both for the model that uses standard NLP features (feature set A) and the model that uses the more refined feature set B in isolation (cf. Table 2). Our models also come close to Friedrich *et al.'s* best results, which they obtain by applying their entire set of features including information from resources like WordNet, with a difference of 2.7 pp accuracy for our best performing model CON_LAB5+GEN. [13]

Attention vectors as input to sequence labeling models. We explored the impact of the attention vectors as inputs to a sequence labeling model – each clause is described through the words with the highest attention weights, and these weights are then used in a conditional random field system (CRF++ [14].). The best performance was obtained when using the attention vector of the target clause (and no additional context) – 61.68% accuracy (47.18% F1 score). CRF++ maps the attention information to binary features, and as such cannot take advantage of information captured in the numerical values of the attention weights, or the embeddings of the given words. Future work includes the development of a CRF that can use continuous values.

Results for single classes. Figure 8 shows macro-average F1 scores of our best performing system CON_LAB5+GEN for the single SE classes. The scores are very similar to the results of Friedrich *et al.* (2016).

13. Since we did not reimplement their system, we cannot report significance results.
14. https://taku910.github.io/crfpp/

Model type		Model Name	Description	Acc	F1
Baselines	**BOW**	BOW+SVM	Bag of Words + SVM	64.83	47.3
	F+16 CRF	CRF, Set A	Standard feature set A	69.8	63.9
		CRF, Set B	Special SE feature set B	71.4	65.5
		CRF, Set A & B	Feature set A & B	**74.7**	**69.3**
Local		LOC	w/o attention	66.55	59.14
		LOC_ATT	with attention	69.18	68.31
		LOC_ATT+GEN	with attention + genre	**71.12**$^{\diamond\triangleleft\square\odot}$	**69.55**
Context	**Tokens**	CON_TOK1+GEN	1 prev. clause + genre	**71.67**$^{\diamond\odot}$	**59.19**
		CON_TOK2+GEN	2 prev. clauses + genre	71.57	48.12
		CON_TOK3+GEN	3 prev. clauses + genre	69.76	42.73
		CON_TOK4+GEN	4 prev. clauses + genre	69.29	41.55
		CON_TOK5+GEN	5 prev. clauses + genre	68.99	30.78
	Labels	CON_LAB1+GEN	1 prev. label + genre	69.55	60.21
		CON_LAB2+GEN	2 prev. labels + genre	71.04	64.54
		CON_LAB3+GEN	3 prev. labels + genre	71.68	64.42
		CON_LAB4+GEN	4 prev. labels + genre	71.25	**65.06**
		CON_LAB5+GEN	5 prev. labels + genre	**72.04**$^{\triangleleft}$	64.74
	Tokens + Labels	CON_TOKLAB1+GEN	1 prev. label/clause + genre	**71.35**$^{\square}$	**70.82**
		CON_TOKLAB2+GEN	2 prev. labels/clauses + genre	70.65	68.62
		CON_TOKLAB3+GEN	3 prev. labels/clauses + genre	69.90	68.83
		CON_TOKLAB4+GEN	4 prev. labels/clauses + genre	69.26	67.47
		CON_TOKLAB5+GEN	5 prev. labels/clauses + genre	69.00	64.36

Table 3. *SE-type classification on English test set. For our models using context information, we only report the results for the best performing models for the following three categories: (i) context models using tokens of previous clauses (*CON_TOK+GEN*); (ii) context models using labels of previous clauses (*CON_LAB+GEN*); and (iii) context models using tokens and labels of previous clauses jointly (*CON_TOKLAB+GEN*). F1 is reported as macro-average score. Significance based on accuracies is computed for the best performing models of each category; pairs of models that are significantly different from each other share the same symbol. Models with the symbol $\odot$ are also signficant with $p < 0.01$.*

Scores for GENERALIZING SENTENCE are the lowest as this class is very infrequent in the data set, while scores for the classes STATE, EVENT, QUESTION and OTHER are the highest. In addition, we explored system performance of CON_LAB5+GEN in a binary (one vs. rest, OvR) classification setting, classifying STATE vs. the remaining classes, EVENT vs. the remaining classes, etc. (cf. Figure 8). Binary classification achieves better performance and can be useful for other tasks which only need information about specific SE-types, for example for distinguishing generic from non-generic sentences. Becker *et al.* (2017) for example showed,

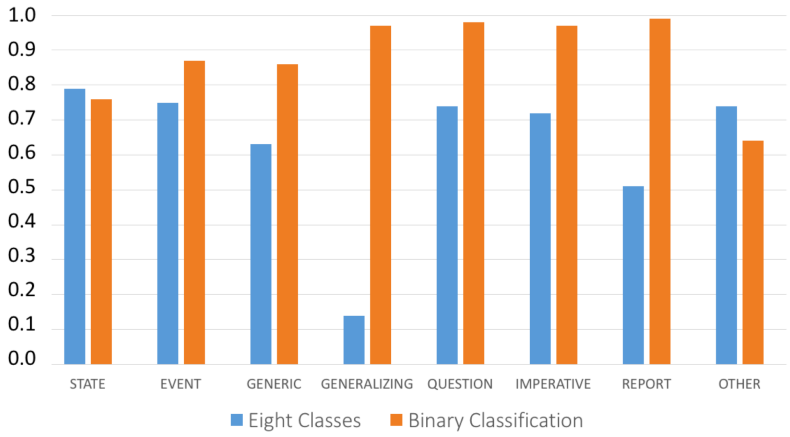


Figure 8. *Macro-average F1 scores of our best performing system* CON_LAB5+GEN *for single SE classes, multiclass vs. binary classification.*

in an annotation experiment of linguistic characteristics of implicit knowledge within argumentative texts, that a majority of implicit information is encoded as GENERIC SENTENCES. This tendency could be deployed for acquiring such knowledge automatically. Other possible application for binary classifications would be extracting non-canonical imperatives or questions for dialogue systems or distinguishing events from non-events as part of systems for event extraction or veridicality determination.

Confusability of classes. Section 5.3 discusses the SE-types with high confusability for human annotators. Most of these same confusions occur with high frequency in the outputs from our best model, as shown in Table 4. In particular, GENERALIZING SENTENCES are often mislabeled as EVENT or STATE – e.g., the clause *Even friendly nations routinely steal information from US companies* is a GENERALIZING SENTENCE but gets labeled as STATE by our model. A major reason for frequently mislabeling GENERALIZING SENTENCES could be that this class is very small within in our dataset (cf. Fig 7). We also find that STATES and GENERIC SENTENCES are frequently confused: e.g., the clause *Certainly the Colombian press is much in need of that* is labeled as GENERIC SENTENCE by our classifier, while the gold label is STATE.

7. Impact and Analysis of Attention and Genre

Our experimental results in Section 6 show that both attention and incorporation of genre information result in improved performance for our local models. In this section, we look more closely at the role of these factors in SE classification.

	Event	Report	State	GenSt	Generic	IMP	Q	Other
Event	**1,255**	25	249	8	58	7	5	115
Report	43	**243**	14	-	-	-	1	16
State	224	6	**2,912**	23	219	31	15	125
GenSt	87	3	109	**40**	58	8	2	19
Generic	70	1	446	12	**948**	8	2	91
IMP	7	-	19	1	7	**165**	1	35
Q	8	1	69	-	3	8	**96**	23
Other	171	2	216	16	117	29	8	**1,306**

Table 4. *Confusion matrix for English of our best performing model* CON_LAB5+GEN.

7.1. *Analysis of Attention*

Attention is an effective mechanism that allows models to focus on specific parts of the input instead of capturing the complete semantics of the input in its hidden state. Beyond this capacity, attention can also give insights into what elements the model learns to be most relevant for predicting the various SE-types. The analyses reported in this section are based on the output of LOC_ATT+GEN, our best performing local model, which is run on the English dataset and uses as input the target clause to be classified jointly with attention and genre information.

We analyze the attention weights learned by the model and focus on different linguistic information: (1) The attention to specific words for specific SE-types; (2) the attention to specific POS tags for specific SE-types and the overall distribution of attention weights among POS tag labels and SE-types; and (3) the position of words with maximum/high attention scores within a clause.

Attention to specific words. When analyzing the characteristics of SE-types regarding words which are assigned high attention scores during training, we find that different classes of words are highlighted for different SE-types. For STATES, nouns and personal pronouns (*youngsters, editors, joyce, I, me*) as well as predicative auxiliaries (*am, are, is*) play a predominant role. In clauses classified as EVENTS, we find many gerunds (*thinking, writing*) with high attention scores, while for GENERIC SENTENCES, adjectives and adverbs (*chronic, awake*), modal verbs (*can, may, must*) and indefinite determiners (*a, an*) are given high attention scores. Interestingly, we find many named entities with high attention scores (*york, states, Miller*) when classifying GENERALIZING SENTENCES. High attention scores for predicative auxiliaries when classifying STATES or for gerunds when classifying EVENTS make sense linguistically. Modal verbs as indicators for GENERIC SENTENCES are very motivated as well, as we often find them with assertions over kinds, such as *Birds can fly, Children must go to school.* However, other findings (e.g., the predominant role of nouns for STATES or of adjectives and adverbs for GENERIC SENTENCES) seem to be arbitrary.

Next, we focus on particular clauses for which adding attention leads to improved classification. Here we analyze attention scores only for those instances that are classified correctly by the attention-enhanced model, while they are incorrectly classified by the model without attention. In these instances, we find many verbs, in particular verbs in past tense (*helped, submitted, included*), which are assigned high attention scores within clauses classified as STATES. Within EVENTS, discourse markers and modifiers (*well, but, some*) are given high attention scores, while in GENERIC SENTENCES modal verbs (*allows, can, must*) play a predominant role. Finally, verba dicendi such as *reported, explained, tells,* or *cited*, get high attention scores when classifying REPORTS, while for the correct classification of QUESTIONS, interrogatives (*what, when, where*) are important. These observations mostly make sense linguistically and highlight the key role of certain word classes.

Attention to specific POS tags. We complement the analysis of attention to specific words with a systematic analysis of attention to POS tags. We therefore post-process our data with POS tags using *spaCy* [15] with the Penn Treebank Tagset (Marcus *et al.*, 1993). Figure 9 visualizes the mean attention score per POS tag for all SE-types (gold labels).

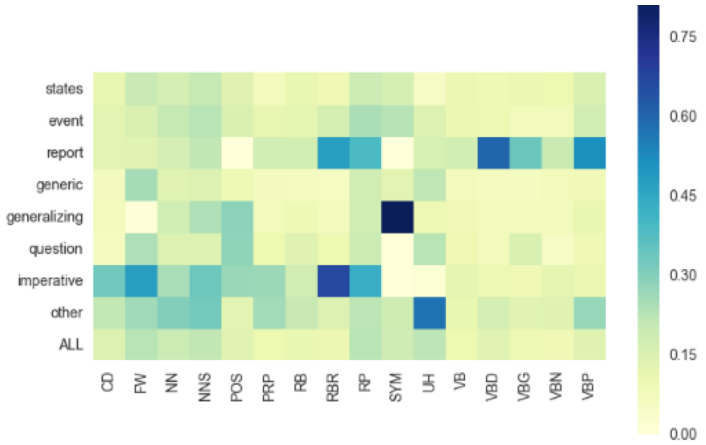


Figure 9. *Mean attention scores for specific SE-types per POS tag on the English training set. POS tags from PTB.*

Interestingly, when we move from focusing on words to focusing on the much broader categories of POS classes, attention weights stand out for classes that are rare, such as IMPERATIVE or REPORT, each less than 5% of the English dataset. Thus, in contrast to sparse lexical material, attention here seems to focus on some more abstract word properties. We don't find outstanding attention weights for particular POS tags when classifying frequent SE-types such as STATE, EVENT or GENERIC SENTENCE.

15. https://spacy.io/docs/usage/pos-tagging.

The heat map indicates that the model attends especially to verbs when classifying the SE-type REPORT. This is not surprising, since REPORT clauses such as *he said* are signaled by verbs of speech. GENERALIZING SENTENCES attend to symbols, mainly punctuation, and genitive markers such as *'s*. The OTHER class, which includes clauses without an assigned SE-type label, attends mostly to interjections. Indeed, OTHER is frequent in genres with fragmented sentences (emails, blogs), and numerous interjections such as *wow* or *um*.

Position of words with high attention scores. Figure 10 shows the relative positions of words with maximum and high attention within clauses. The model mostly attends to words at the end of clauses and almost never to words in the first half of clauses. This distribution shifts to the left when considering more words with high attention scores instead of only the word with maximum attention – words with 2^{nd} (3^{rd}, 4^{th}, 5^{th}) highest attention score can often be found at the beginning of clauses. Thus, the model seems to draw information from a broad range of positions.

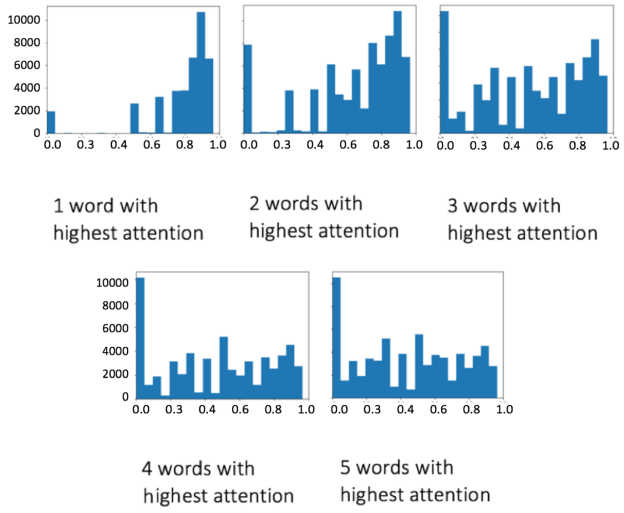


Figure 10. *Position of words with maximum attention within clauses; x-axis represents the normalized position within the clause, y-axis the number of words with maximum attention at that position.*

The analysis of attention weights yields a number of expected findings, mostly for easily-characterized (though infrequent) classes such as REPORT and QUESTION. For more frequent and more varied classes such as EVENT, STATE, and GENERIC SENTENCE, neither single words nor single POS tags seem to provide especially strong signals for SE classification. Analysis of attention and word position indeed suggests that the model attends to multiple elements of the clause in order to arrive at an SE-label.

7.2. *Analysis of Genre*

We investigate more deeply the relevance of genre information on the classification performance and to what extent genre information is reflected in SE-label sequences. [16]

We first compare the accuracy of the best performing local model and the best performing context model, respectively with and without genre information (LOC_ATT+GEN vs. LOC_ATT and CON_LAB5+GEN vs. CON_LAB5), for the English dataset. The results are given in Figure 11. For some genres (e.g., news, ficlets, and emails), LOC_ATT performs quite well and shows little benefit from the inclusion of genre information. For governmental protocols, technical reports, Wikipedia articles, travel reports, and letters, on the other hand, LOC_ATT benefits quite a bit from genre information (i.e., LOC_ATT+GEN far outperforms LOC_ATT). CON_LAB5, which uses context in form of the labels of previous clauses, in general seems not to benefit as much from genre information as LOC_ATT (while performing better overall due to context information). Figure 11 shows for example that genres such as Ficlets, Emails, Fiction or Journal articles benefit very little or not at all from the inclusion of genre information, while governmental protocols and letters benefit from genre information.

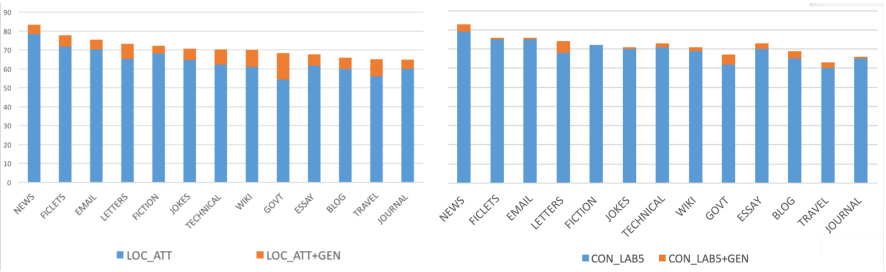


Figure 11. *Comparing performances (accuracy) of our best performing local model* LOC_ATT *and our best performing context model* CON_LAB5*, respectively with and without genre information, for the genres of our English test set separately.*

In order to further analyze the effects of genre information on SE-type classification, we explore the similarity of genres with respect to the distribution of SE-types

16. We didn't train the classifier strictly within specific genres mainly for two reasons: The first reason is that we have too little data for some genres such as technical reports, letters or essays (see Figures 11 and 12), and even for the genres with a comparably high number of instances such as Wikipedia, the data size is still quite low to train our system sufficiently without overfitting. But even more important, one crucial aim in developing the classifier was to build a system which is robust across genres when classifying SE-types, which highlights the importance of training our model across various genres at the same time.

and their sequences (modeled as bigrams) measured by symmetric Kullback-Leibler divergence:

$$D_{klsym}(P||Q) = D_{KL}(P||Q) + D_{KL}(Q||P) \quad [12]$$

where

$$D_{KL}(P||Q) = \sum_i P(i) * \log \frac{P(i)}{Q(i)} \quad [13]$$

P and Q are the corresponding distributions of SE-types (unigrams or bigrams) for different genres, whereas i iterates over all possible outcomes. The results based on unigrams and bigrams of SE-types are visualized as a heat map in Figure 12. News and Wikipedia articles as well as news and emails show high values and therefore differ a lot with respect to the distribution of both uni- and bigrams of SE-types, while jokes and blog articles or journal articles and fiction are more similar. For some genres (Wikipedia articles in particular) the findings from this analysis correspond to the size of performance improvement due to incorporation of genre information (cf. Figure 11). For others (e.g., news and emails) the correspondence doesn't hold.

Figure 12. *Distributional divergence of SE-types across genres: symmetric Kullback-Leibler divergence of unigrams (left) and bigrams (right) of SE-types within the English dataset (train+dev+test set).*

Overall, bigrams show larger differences than unigrams. Wikipedia articles, followed by travel reports, show the highest values both for unigram and bigram analyses. We also see that Wikipedia articles and travel reports are both dissimilar to essays, ficlets, fiction, and news, while essays, ficlets, and fiction are on the other hand quite similar to government documents (light green). This suggests that the improvements in Wikipedia articles and travel reports are strongly related to each other, and that these genres profit mutually from the genre information given to the model. The distributional similarities and dissimilarities seem to be exploited in the models using genre information. We can also expect that genres that are sparse can profit from genres with

Model type		Name	Description	Acc	F1
Local		LOC	w/o attention	74.94	67.12
		LOC_ATT	with attention	74.51	**74.02**
		LOC_ATT+GEN	with attention + genre	**75.56**$^{\diamond\triangleleft\square}$	69.98
Context	**Tokens**	CON_TOK1+GEN	1 prev. clause + genre	**74.51**$^{\diamond}$	**72.41**
		CON_TOK2+GEN	2 prev. clauses + genre	74.44	72.26
		CON_TOK3+GEN	3 prev. clauses + genre	73.35	71.79
		CON_TOK4+GEN	4 prev. clauses + genre	73.11	71.12
		CON_TOK5+GEN	5 prev. clauses + genre	72.89	70.61
	Labels	CON_LAB1+GEN	1 prev. label + genre	71.78	**52.88**
		CON_LAB2+GEN	2 prev. labels + genre	72.29	52.52
		CON_LAB3+GEN	3 prev. labels + genre	72.47	52.34
		CON_LAB4+GEN	4 prev. labels + genre	74.33	51.12
		CON_LAB5+GEN	5 prev. labels + genre	**74.92**$^{\triangleleft}$	50.76
	Tokens + Labels	CON_TOKLAB1+GEN	1 prev. label/clause + genre	**73.43**$^{\square}$	**59.51**
		CON_TOKLAB2+GEN	2 prev. labels/clauses + genre	72.23	57.38
		CON_TOKLAB3+GEN	3 prev. labels/clauses + genre	71.69	57.99
		CON_TOKLAB4+GEN	4 prev. labels/clauses + genre	71.11	56.48
		CON_TOKLAB5+GEN	5 prev. labels/clauses + genre	71.09	56.23

Table 5. *SE-type classification on German test set. Again, we only report the results for the best performing models for the context models (*CON_TOK+GEN*,* CON_LAB+GEN *and* CON_TOKLAB+GEN*). F1 is reported as macro-average score. Pairs of models that yield significant performance differences are marked with the same symbol; significance is computed for the best performing models of each category.*

similar SE-type distributions that are more frequent (cf. Figure 6). In future work, it would be interesting to consider other approaches to measuring genre similarity, such as overlap of lexical items, syntactic structures, or topic model distributions.

8. Porting the System to German

A great advantage of neural-based systems is that they are able to learn relevant features for classification during the training procedure, and thus do not rely on hand-crafted features. This is of considerable help when models are to be transferred to novel languages, when such features are expensive to compute, or not available because of lack of resources.

We use the system described above with German data, and adjust the size of the input embeddings. [17] We tune hyperparameters separately for German on the German development set through random search using *optunity* (Bergstra and Bengio, 2012) and use the best setting for evaluating on the test set. As for the English dataset, we tune batch size, number of layers, GRU cell size, and regularization parameter (L2), we use AdaGrad (Duchi *et al.*, 2011) for learning rate optimization, and we tune the initial learning rate. For LOC, the best result on the development set is achieved for GRU with batch size 100, 2 layers, cell size 124, learning rate 0.05, and L2 regularization parameter (0.01). For LOC_ATT the parameters are identical, except for L2 (0.0001). As for English, we again apply the optimal hyperparameters for LOC_ATT to the local model LOC_ATT+GEN and to the context models CON_TOK+GEN, CON_LAB+GEN and CON_LABTOK+GEN.

Table 5 gives an overview of the results for different models, and allows us to compare the effectiveness of integrating context and genre information. Compared to English, the local models achieve higher performance, but attention by itself does not improve the results (cf. LOC vs. LOC_ATT). Used jointly, attention and genre information LOC_ATT+GEN yield a moderate increase of 0.62 pp accuracy compared to LOC. Attention may need more data and possibly more diversity to be learned effectively. Improving the attention models for German will be the focus of our future work, to facilitate a meaningful linguistic analysis of attention for German.

In the case of German, modeling context information doesn't improve results: compared to the best local model (LOC_ATT+GEN), adding more context either in form of the tokens (CON_TOK+GEN) or labels of previous clauses (CON_LAB+GEN) or both (CON_TOKLAB+GEN) does not lead to higher accuracy (cf. Table 5). Again, this can be due to the smaller dataset size, which may not provide enough data points for the richer context models.

9. Conclusion

We presented an RNN-based approach to SE-type classification that bears clear advantages compared to previous classifier models that rely on sophisticated, hand-engineered features and lexical semantic resources: given pre-annotated training data, our neural model is easily transferable to other languages as it can tune pre-trained word embeddings to encode semantic information relevant for the task.

We designed and compared several GRU-based RNN models that jointly model *local and contextual* information in a unified architecture. Genre information was added to exploit common properties of specific textual genres. What makes our work interesting for linguistically informed semantic models is the exploration of different model variants that combine local classification with sequence information gained

17. The different size of the embeddings (for English and German cf. Section 4.4) may have an impact on the results.

from the contextual history, and the analysis of the interaction between these properties and genre characteristics as well as the interaction of sequence information and genre.

Our best model trained on English data jointly uses genre and context information in the form of previously predicted labels and is enhanced with attention. It outperforms the state-of-the-art models of Friedrich *et al.* (2016) for English when using either off-the-shelf NLP features (set A) or, separately, hand-crafted features based on lexical resources (set B). A small margin of less than 3 pp accuracy is left to achieve in future work to compete with the knowledge-rich model combining both feature sets.

For the German models we find that the local model enhanced with attention and genre information leads to highest accuracies, while modeling context information doesn't improve the results. We leave improving the attention and context models for German as future work.

For our English dataset, we show that, by using attention, we can gain insights into what the models learn. The analysis of attention weights shows interesting findings, especially for easily characterized classes such as REPORT and QUESTION, while for more varied classes such as EVENT, STATE, and GENERIC SENTENCE, we don't observe clear patterns or distributions regarding single words or POS tags (which could be helpful for classification), notwithstanding the fact that the attention mechanism in general improves our models. Some of our findings can be motivated linguistically, e.g., high attention scores for predicative auxiliaries when classifying STATES or modal verbs as indicators for GENERIC SENTENCES. We further observe that our models attend to multiple elements of the clause during training and therefore seem to draw information from a broad range of positions within clauses.

Our analyses of the impact of genre information (again on the English dataset) show that genre improves classification performance across the board, but some genres benefit more from this information than others, which can be partially linked to variation in SE-type distributions. Our models can be used either as multi-class or as binary classifiers for detecting events, generics, imperatives, or questions; they can help model discourse modes or can improve argument analysis and argument detection tasks.

Acknowledgments. We thank Sabrina Effenberger, Jesper Klein, Sarina Meyer, and Rebekka Sons for the annotations and their helpful feedback on the annotation manual. This research is funded by the Leibniz Science Campus Empirical Linguistics & Computational Language Modeling, supported by Leibniz Association grant no. SAS-2015-IDS-LWC and by the Ministry of Science, Research, and Art of Baden-Württemberg. We also acknowledge a grant from NVIDIA Corporation.

10. References

Abdul-Mageed M., Ungar L., "EmoNet: Fine-Grained Emotion Detection With Gated Recurrent Neural Networks", *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 718-728, 2017.

Asher N., *Reference to Abstract Objects in Discourse*, Kluwer Academic Publishers, 1993.

Bahdanau D., Cho K., Bengio Y., "Neural Machine Translation by Jointly Learning to Align and Translate", *International Conference on Machine Learning*, 2015.

Becker M., Palmer A., Frank A., "Argumentative Texts and Clause Types", *Proceedings of the 3rd Workshop on Argument Mining*, p. 21-30, 2016a.

Becker M., Palmer A., Frank A., "Clause Types and Modality in Argumentative Microtexts", *Workshop on Foundations of the Language of Argumentation (in conjunction with COMMA)*, p. 1-9, 2016b.

Becker M., Staniek M., Palmer A., Nastase V., Frank A., "Classifying Semantic Clause Types: Modeling Context and Genre Characteristics with Recurrent Neural Networks and Attention", *Proceedings of the Joint Conference on Lexical and Computational Semantics (Starsem)*, p. 230-240, 2017.

Bengio Y., Simard P., Frasconi P., "Learning Long-term Dependencies with Gradient Descent is Difficult", *IEEE Transactions of Neural Networks*, vol. 5, n° 2, p. 157-166, 1994.

Bergstra J., Bengio Y., "Random Search for Hyper-Parameter Optimization", *Journal of Machine Learning Research*, vol. 13, p. 281-305, 2012.

Brown P. F., Desouza P. V., Mercer R. L., Pietra V. J. D., Lai J. C., "Class-Based N-gram Models of Natural Language", *Computational Linguistics*, vol. 18, n° 4, p. 467–479, 1992.

Carlson G. N., Pelletier F. J. (eds), *The Generic Book*, University of Chicago Press, 1995.

Cheng F., Miyao Y., "Classifying Temporal Relations by Bidirectional LSTM over Dependency Paths", *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 1-6, 2017.

Cho K., van Merrienboer B., Bahdanau D., Bengio Y., *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*, 8th Workshop on Syntax, Semantics and Structure in Statistical Translation, 2014.

Conneau A., Kiela D., Schwenk H., Barrault L., Bordes A., "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data", *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, p. 670-680, 2017.

Costa F., Branco A., "Aspectual Type and Temporal Relation Classification", *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 266-275, 2012.

Dai Z., Huang R., "Building Context-aware Clause Representations for Situation Entity Type Classification", *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, p. 3305-3315, 2018.

Dong L., Lapata M., "Language to Logical Form With Neural Attention", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 33-43, 2016.

Dowty D., *Word Meaning and Montague Grammar*, Reidel, 1979.

Duchi J., Hazan E., Singer Y., "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", *Journal of Machine Learning Research*, vol. 12, nº 7, p. 2121-2159, 2011.

Francis N., Kucera H., "A Standard Corpus of Present-Day Edited American English, for Use with Digital Computers.", *Department of Linguistics, Brown University, Providence, Rhode Island, USA*, 1979.

Friedrich A., States, Events, and Generics: Computational Modeling of Situation Entity Types, PhD thesis, Universität des Saarlandes, 2017.

Friedrich A., Palmer A., "Automatic Prediction of Aspectual Class of Verbs in Context", *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 517-523, 2014a.

Friedrich A., Palmer A., "Situation Entity Annotation", *Proceedings of the Linguistic Annotation Workshop VIII*, p. 149-158, 2014b.

Friedrich A., Palmer A., Pinkal M., "Situation Entity Types: Automatic Classification of Clause-Level Aspect", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, p. 1757-1768, 2016.

Friedrich A., Palmer A., Sørensen M. P., Pinkal M., "Annotating Genericity: a Survey, a Scheme, and a Corpus", *The 9th Linguistic Annotation Workshop held in conjuncion with NAACL 2015*, p. 21, 2015.

Friedrich A., Pinkal M., "Discourse-sensitive Automatic Identification of Generic Expressions", *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 1272-1281, 2015.

Goldberg Y., *Neural Network Methods for Natural Language Processing*, vol. 37 of *Synthesis Lectures on Human Language Technologies*, Morgan & Claypool, San Rafael, CA, 2017.

Goodfellow I., Bengio Y., Courville A., *Deep Learning*, MIT Press, 2016. `http://www.deeplearningbook.org`.

Herbelot A., Copestake A., "Annotating Genericity: How Do Humans Decide? (A Case Study in Ontology Extraction)", *Studies in Generative Grammar*p. 103, 2009.

Hochreiter S., Schmidhuber J., "Long Short-Term Memory", *Neural computation*, vol. 9, nº 8, p. 1735-1780, 1997.

Ide N., Fellbaum C., Baker C., Passonneau R., "The Manually Annotated Sub-Corpus: A Community Resource For and By the People", *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 68-73, 2010.

Joulin A., Grave E., Bojanowski P., Mikolov T., "Bag of Tricks for Efficient Text Classification", *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics,*p. 427-431, 2017.

Kalchbrenner N., Grefenstette E., Blunsom P., "A Convolutional Neural Network for Modelling Sentences", *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Baltimore, Maryland, p. 655-665, 2014.

Kim Y., "Convolutional Neural Networks for Sentence Classification", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, p. 1746-1751, 2014.

Lample G., Ballesteros M., Subramanian S., Kawakami K., Dyer C., "Neural Architectures for Named Entity Recognition", *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, p. 260-270, 2016.

Loaiciga S., Meyer T., Popescu-Belis A., "English-French Verb Phrase Alignment in Europarl for Tense Translation Modeling", *Proceedings of the 9th Language Resources and Evaluation Conference (LREC)*, 2014.

Marcus M., Santorini B., Marcinkiewicz M. A., "Building a Large Annotated Corpus of English: The Penn Treebank", *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1993.

Mavridou K.-I., Friedrich A., Sorensen M., Palmer A., Pinkal M., "Linking Discourse Modes and Situation Entities in a Cross-Linguistic Corpus Study", *Proceedings of the EMNLP Workshop LSDSem 2015: Linking Models of Lexical, Sentential and Discourse-level Semantics*, 2015.

Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., "Distributed Representations of Words and Phrases and Their Compositionality", *Advances in neural information processing systems*, p. 3111-3119, 2013a.

Mikolov T., Yih W.-t., Zweig G., "Linguistic Regularities in Continuous Space Word Representations", *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, p. 746-751, 2013b.

Mishra A., Dey K., Bhattacharyya P., "Learning Cognitive Features from Gaze Data for Sentiment and Sarcasm Classification using Convolutional Neural Network", *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 377-387, 2017.

Miwa M., Bansal M., "End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 1105-1116, 2016.

Nedoluzhko A., "Generic Noun Phrases and Annotation of Coreference and Bridging Relations in the Prague Dependency Treebank", *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, p. 103-111, 2013.

Nie A., Bennett E. D., Goodman N. D., "DisSent: Sentence Representation Learning from Explicit Discourse Relations", *Computing Research Repository*, 2017.

O'Seaghdha D., Teufel S., "Unsupervised Learning of Rhetorical Structure with Un-Topic Models", *Proceedings of the 25th International Conference on Computational Linguistics*, Association for Computational Linguistics, p. 2-13, 2014.

Palmer A., Friedrich A., "Genre Distinctions and Discourse Modes: Text Types Differ in Their Situation Type Distributions", *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and NLP*, 2014.

Palmer A., Ponvert E., Baldridge J., Smith C., "A Sequencing Model for Situation Entity Classification", *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 896-903, 2007.

Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M.,

Perrot M., Duchesnay E., "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, vol. 12, p. 2825-2830, 2011.

Peldszus A., Stede M., "An Annotated Corpus of Argumentative Microtexts", *Proceedings of the First European Conference on Argumentation*, 2015.

Plank B., Søgaard A., Goldberg Y., "Multilingual Part-of-Speech Tagging With Bidirectional Long Short-Term Memory Models and Auxiliary Loss", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 412-418, 2016.

Reimers N., Eckle-Kohler J., Schnober C., Kim J., Gurevych I., "Germeval-2014: Nested Named Entity Recognition with Neural Networks", *Proceedings of the 12th Edition of the KONVENS Conference*, p. 117–120, 2014.

Reiter N., Frank A., "Identifying Generic Noun Phrases", *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, p. 40-49, 2010.

Rocktäschel T., Grefenstette E., Hermann K. M., Kočiskỳ T., Blunsom P., "Reasoning About Entailment With Neural Attention", *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, May, 2016.

Sanagavarapu K. C., Vempala A., Blanco E., "Determining Whether and When People Participate in the Events They Tweet About", *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 641-646, 2017.

Sidarenka U., Peldszus A., Stede M., "Discourse Segmentation of German Texts", *Journal for Language Technology and Computational Linguistics*, vol. 30, p. 71-98, 2015.

Siegel E. V., McKeown K. R., "Learning Methods to Combine Linguistic Indicators: Improving Aspectual Classification and Revealing Linguistic Insights", *Computational Linguistics*, vol. 26, nº 4, p. 595-628, 2000.

Smith C. S., *The Parameter of Aspect*, Kluwer, 1991.

Smith C. S., *Modes of Discourse: The Local Structure of Texts*, vol. 103, Cambridge University Press, 2003.

Smith C. S., "Aspectual Entities and Tense in Discourse", *Aspectual inquiries*, Springer, p. 223-237, 2005.

Song W., Wang D., Fu R., Liu L., Liu T., Hu G., "Discourse Mode Identification in Essays", *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 112-122, 2017.

Soricut R., Marcu D., "Sentence Level Discourse Parsing Using Syntactic and Lexical Information", *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology.*, Association for Computational Linguistics, p. 149-15, 2003.

Tai K. S., Socher R., Manning C. D., "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks", *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, p. 1556-1566, 2015.

Teufel S., Argumentative Zoning: Information Extraction From Scientific Text, PhD thesis, University of Edinburgh, 2000.

Theano Development Team, "Theano: A Python Framework for Fast Computation of Mathematical Expressions", *arXiv e-prints*, May, 2016.

Vempala A., Blanco E., Palmer A., "Determining Event Durations: Models and Error Analysis", *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, p. 164-168, 2018.

Vendler Z., "Verbs and Times", *The Philosophical Review,*p. 143-160, 1957.

Verkuyl H., *On the Compositional Nature of the Aspects*, Reidel, 1972.

Vu N. T., Adel H., Gupta P., Schütze H., "Combining Recurrent and Convolutional Neural Networks for Relation Classification", *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 534-539, 2016.

Wang Y., Huang M., zhu x., Zhao L., "Attention-based LSTM for Aspect-level Sentiment Classification", *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, p. 606-615, 2016.

Yin W., Kann K., Yu M., Schütze H., "Comparative Study of CNN and RNN for Natural Language Processing", *Computing Research Repository*, 2017.

Zarcone A., Lenci A., "Computational Models of Event Type Classification in Context", *Proceedings of the International Conference on Language Resources and Evaluation*, 2008.

Zhou J., Cao Y., Wang X., Li P., Xu W., "Deep Recurrent Models With Fast-Forward Connections for Neural Machine Translation", *Transactions of the Association for Computational Linguistics,*p. 371-383, 2016.

Zhou J., Xu W., "End-to-end Learning of Semantic Role Labeling Using Recurrent Neural Networks", *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Association for Computational Linguistics, p. 1127-1137, 2015.

Prédiction de performance des systèmes de reconnaissance automatique de la parole à l'aide de réseaux de neurones convolutifs

Zied Elloumi[1,2], **Benjamin Lecouteux**[2], **Olivier Galibert**[1], **Laurent Besacier**[2]

[1] *Laboratoire national de métrologie et d'essais (LNE) , France*
prénom.nom@univ-grenoble-alpes.fr

[2] *Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France*
prénom.nom@lne.fr

RÉSUMÉ. *Dans ce travail, nous nous intéressons à la tâche de prédiction de performance des systèmes de transcription de la parole. Nous comparons deux approches de prédiction: une approche de l'état de l'art fondée sur l'extraction explicite de traits et une nouvelle approche fondée sur des caractéristiques entraînées implicitement à l'aide des réseaux neuronaux convolutifs (CNN). Nous essayons ensuite de comprendre quelles informations sont capturées par notre modèle neuronal et leurs liens avec différents facteurs. Pour tirer profit de cette analyse, nous proposons un système multitâche qui se montre légèrement plus efficace sur la tâche de prédiction de performance.*

ABSTRACT. *This paper focuses on the ASR performance prediction task. Two prediction approaches are compared: a state-of-the-art performance prediction based on engineered features and a new strategy based on learnt features using convolutional neural networks. We also try to better understand which information is captured by the deep model and its relation with different conditioning factors. To take advantage of this analysis, we then try to leverage these 3 types of information at training time through multi-task learning, which is slightly more efficient on ASR performance prediction task.*

MOTS-CLÉS : *prédiction de performance, reconnaissance de la parole continue à grand vocabulaire, réseau neuronal convolutif.*

KEYWORDS: *performance prediction, large vocabulary continuous speech recognition, convolutional neural networks.*

1. Introduction

Prédire la performance d'un système de reconnaissance automatique de la parole (SRAP) sur de nouveaux enregistrements (par exemple de nouveaux types de programmes TV ou radio jamais rencontrés auparavant) est un Graal important de la reconnaissance automatique de la parole. Résoudre une telle tâche permet de prévoir la difficulté de transcription d'une nouvelle collection de documents audio et d'avoir une idée de l'effort nécessaire qui sera demandé à des annotateurs humains pour produire des transcriptions (références) correctes à partir de transcriptions automatiques (hypothèses).

Un système de prédiction de performance prend en entrée des données textuelles (l'hypothèse du système de SRAP) et/ou acoustiques (le signal source ayant donné l'hypothèse) et prédit une performance (un taux d'erreurs de mots ou *word error rate*) associée sans disposer de la transcription de référence. Ceci peut être fait à différentes granularités : tour de parole, document, type de programme TV ou radio, etc. De plus, pour la tâche de prédiction de performance, le système de reconnaissance de la parole est généralement considéré comme une « boîte noire », c'est-à-dire un système pour lequel nous n'aurons accès au fonctionnement interne, ni aux N meilleures hypothèses.

La prédiction de performance de transcription automatique à partir d'une collection de signaux est légèrement différente de la détection d'erreurs ou de l'estimation de mesures de confiance sur une sortie de système de reconnaissance de la parole (SRAP) car elle a pour but de donner une estimation générale de la difficulté de la tâche de transcription sur un ensemble d'enregistrements. Par ailleurs, résoudre une telle tâche de prédiction peut aider à mieux analyser les difficultés rencontrées par les systèmes de SRAP sur différents types d'enregistrements et les facteurs qui affectent leurs performances.

1.1. *Contribution*

Dans ce travail, nous nous intéressons à la tâche de prédiction de performance des systèmes de reconnaissance de la parole (SRAP) sur des collections d'émissions TV ou radio. Cet article reprend des résultats déjà publiés en anglais (Elloumi *et al.*, 2018) où nous proposons un corpus ainsi qu'un protocole d'évaluation pour la prédiction de performance de SRAP associés à des expérimentations fondées sur deux approches de prédiction des performances fondées sur des traits explicites (*engineered features*) et sur des traits entraînés au cours de l'apprentissage de réseaux de neurones convolutifs (*learnt features*). Nous y ajoutons une contribution originale visant à analyser les représentations apprises et les informations capturées par le réseau de neurones.

Nous commençons par présenter un corpus en français large et hétérogène (multiples programmes TV ou radio, mélange de la parole non spontanée et spontanée, différents accents) dédié à cette tâche, ainsi que le protocole d'évaluation utilisé. Nous comparons deux approches de prédiction : une approche de l'état de l'art fon-

dée sur des traits explicites et une nouvelle approche fondée sur des réseaux neuronaux convolutifs (CNN). L'utilisation jointe de traits textuels et acoustiques n'apporte pas de gains dans l'approche de l'état de l'art, tandis qu'elle permet d'obtenir de meilleures prédictions en utilisant les CNN. Nous montrons également que les CNN prédisent clairement la distribution des taux d'erreurs sur une collection d'enregistrements, contrairement à l'approche de l'état de l'art qui génère une distribution éloignée de la réalité. Nous essayons ensuite de comprendre quelles informations sont capturées par notre modèle neuronal et leurs liens avec différents facteurs (style de parole, accent, etc.). Nos expériences montrent que les représentations intermédiaires dans le réseau encodent spontanément des informations sur le style de la parole, l'accent du locuteur ainsi que le type d'émission. Pour tirer profit de cette analyse, nous proposons une approche multitâche qui se montre légèrement plus efficace sur la tâche de prédiction de performance.

1.2. *Plan*

Ce document est organisé comme suit : dans un premier temps, nous présentons dans la section 2 les travaux existants sur la tâche de prédiction de performance ainsi que sur l'analyse des représentions intermédiaires apprises par les réseaux de neurones. Dans la section 3, nous présentons notre protocole d'évaluation. Nous comparons par la suite dans la section 4 deux approches de prédiction (TranscRater *vs* CNN). Dans la section 5, nous détaillons la méthodologie utilisée pour analyser et évaluer des représentations intermédiaires apprises par notre meilleur système neuronal profond. Ensuite, nous présentons, dans la section 6, les systèmes de prédiction multitâche qui exploitent, au cours de l'apprentissage, des informations telles que style de parole, accent du locuteur et type d'émission. Finalement, nous concluons notre travail dans la section 7.

2. Travaux liés

De nombreux travaux ont proposé d'estimer des mesures de confiance afin de détecter les erreurs dans les sorties des SRAP. La détection des erreurs consiste à étiqueter chaque mot en entrée comme « correct » ou « incorrect » (tâche de classification). Les mesures de confiance ont été introduites pour la tâche de détection des mots hors vocabulaire (OOV) par Asadi *et al.* (1990) et exploitées par Young (1994) qui a utilisé les probabilités *a posteriori* (WPP) pour la tâche de reconnaissance de la parole.

La tâche de prédiction des performances va au-delà de l'estimation de confiance puisqu'elle ne se concentre pas sur un système de reconnaissance automatique de la parole donné ni sur des treillis ou des N meilleures hypothèses. Elle a pour but de donner une estimation générale de la difficulté de la tâche de transcription. Tandis que la tâche de prédiction de performance est traitée à l'aide d'approches à base de régression, la tâche d'estimation de confiance est traitée avec des approches à base de classification. Plusieurs travaux se fondent essentiellement sur des traits acoustiques

pour prédire les performances, Hermansky *et al.* (2013) exploitent des caractéristiques temporelles du signal vocal (*Mean Temporal Distance* – calculées sur le signal et corrélées avec le rapport signal sur bruit) pour prédire la performance. Ferreira *et al.* (2018) proposent d'analyser le comportement de l'énergie à court terme du bruit et de la parole en tenant compte de divers facteurs tandis que le système RAP est considéré comme une boîte noire. Les auteurs comparent deux approches de régression (MLP et linéaire) en prenant en compte la variabilité des systèmes de RAP en fonction du volume et du type de bruit. Les performances obtenues montrent que la régression MLP est meilleure que la régression linéaire. Meyer *et al.* (2017) proposent une méthode de prédiction de performance de RAP apprise avec des données propres et évaluée sur 10 types de bruits inconnus et une large gamme de rapports signal/bruit des corpus DRE01 (Dreschler *et al.*, 2001), Noisex et BBC SOUND EFFECTS. Les résultats montrent que le bruit dans les données influence la qualité des systèmes de prédiction. Negri *et al.* (2014) proposent d'autres types de traits (autres que le signal) comme les informations internes d'un SRAP, des caractéristiques acoustiques, des caractéristiques hybrides, et des caractéristiques textuelles. Trois scénarios de prédiction ont été proposés afin d'étudier l'impact de présence/absence de caractéristiques particulières extraites des SRAP, ainsi que l'effet de l'homogénéité/non-homogénéité des données d'apprentissage et d'évaluation sur la qualité des systèmes de prédiction. Les performances obtenues montrent que la qualité des systèmes de prédiction dépend de l'homogénéité entre les données d'entraînement et les données d'évaluation. Jalalvand *et al.* (2016) ont proposé un outil *open source* nommé TranscRater qui se fonde essentiellement sur l'extraction de traits (caractéristiques phonétiques, syntaxiques, acoustiques et issues du modèle de langue) et utilise un algorithme fondé sur une régression pour prédire un taux d'erreurs. Le SRAP est considéré comme une « boîte noire », et l'évaluation a été effectuée sur les données de la campagne CHiME-3. [1] Dans ce travail, qui se rapproche le plus de notre contribution, les expérimentations montrent que les caractéristiques acoustiques (issues directement du signal) n'ont pas d'influence sur la qualité du système de prédiction.

Les travaux présentés précédemment s'appuient sur des traits (ou *features*) prédéfinis qui exigent des outils et des ressources spécifiques pour une langue afin de prédire la performance. À l'aide des CNN, nous visons à proposer une méthode flexible (ne dépendant pas de la langue) qui se fonde sur des représentations apprises au cours de l'apprentissage du système. Un autre apport de notre travail est d'encoder les informations du signal de parole dans un CNN pour la prédiction de performance des SRAP. L'encodage du signal pour un CNN peut être effectué à partir de la sortie d'un module de traitement du signal (*front end*) qui transforme le signal en une suite de vecteurs acoustiques (Piczak, 2015 ; Sainath *et al.*, 2015 ; Jin *et al.*, 2016). Cependant, certains travaux récents ont directement utilisé le signal brut en entrée d'un CNN pour la reconnaissance vocale (Sainath *et al.*, 2015 ; Palaz *et al.*, 2015) et pour la classification de signaux de parole (Dai *et al.*, 2017). Des travaux liés à cet objectif ont aussi

1. `http://spandh.dcs.shef.ac.uk/chime_challenge/chime2015/`

consisté à détecter des classes phonétiques à partir de signaux de parole analysés avec des réseaux neuronaux (Pellegrini et Mouysset, 2016 ; Nagamine *et al.*, 2015).

La contribution principale présentée ici est la prédiction de performance à l'aide de réseaux de neurones qui se fondent sur des traits multimodaux : acoustiques et textuels observés au cours de l'apprentissage. Toutefois, il est important d'interpréter les représentations intermédiaires apprises par le réseau afin de comprendre quelles informations ont été capturées. Des travaux récents sur la tâche de reconnaissance automatique de la parole ont proposé d'analyser les représentations capturées par les SRAP profonds. Mohamed *et al.* (2012) et Belinkov et Glass (2017) ont analysé les représentations intermédiaires apprises (d'un SRAP profond) en utilisant la visualisation t-SNE (Maaten et Hinton, 2008). Ils essaient aussi de comprendre quelles couches capturent mieux les informations phonétiques en entraînant un classifieur de phonèmes peu profond. Par ailleurs, Wu et King (2016) ont évalué les représentations de plusieurs variantes de LSTM pour une tâche de synthèse vocale. Wang *et al.* (2017) ont, quant à eux, proposé une étude sur trois types de représentations apprises pour une tâche de reconnaissance de locuteur : i-vecteur, d-vecteur et s-vecteur (fondé sur un réseau LSTM). Des tâches de classification annexes ont été conçues pour mieux comprendre comment sont encodées les informations sur les locuteurs. Un apprentissage multitâche est également proposé pour intégrer ces différents types de représentations, ce qui mène à une meilleure performance d'identification du locuteur. Nous trouvons aussi des travaux similaires dans d'autres applications du traitement automatique du langage naturel (TALN), comme en traduction automatique neuronale par exemple. Parmi ces travaux récents, nous pouvons citer les travaux de Shi *et al.* (2016) et Belinkov *et al.* (2017) qui ont essayé de comprendre les représentations apprises par un système de traduction neuronal. Ces représentations sont fournies à un classifieur peu profond afin de prédire des étiquettes syntaxiques (Shi *et al.*, 2016), grammaticales ou sémantiques (Belinkov *et al.*, 2017). L'analyse montre que les couches inférieures sont meilleures pour l'étiquetage grammatical.

3. Protocole d'évaluation

Nous nous intéressons à la prédiction des performances de systèmes de transcription de la parole sur des émissions non vues durant l'apprentissage. Notre objectif est de prédire la performance lorsque les informations internes d'un SRAP sont indisponibles. Notre cas d'étude se fonde sur un système de prédiction de performance n'utilisant que les transcriptions automatiques (fournies par un SRAP) et/ou le signal audio afin de prédire la performance de la transcription correspondante. Les transcriptions de référence (humaines) ne sont disponibles que pour évaluer le système de transcription de la parole et pour produire un rapport de performance. Un corpus nommé $\textit{Train}_{pred}$ est utilisé pour construire nos systèmes de prédiction. Il est constitué de triplets {signaux, transcription automatique, performance} pour 75 k tours de parole. Le corpus $\textit{Test}_{pred}$ est utilisé pour évaluer le système de prédiction, il contient aussi les triplets {signaux, transcription automatique, performance} (6,8 k tours de

parole) mais la performance est inconnue du système au moment de la prédiction et dévoilée seulement au moment de l'évaluation de la qualité de prédiction.

Afin d'implémenter ce protocole, nous avons besoin d'un système de reconnaissance de la parole pour produire les transcriptions automatiques et la performance associée pour l'intégralité des corpus Train$_{pred}$ et Test $_{pred}$, ce qui nous permettra d'entraîner et d'évaluer des systèmes de prédiction de performance.

3.1. *Corpus*

Les données utilisées dans notre protocole proviennent de différentes collections d'émissions en français :

1) un sous-ensemble du corpus Quaero[2] qui contient 41 heures de discours radiodiffusés de différents programmes de radio et de télévision français sur divers sujets ;

2) les données du projet ETAPE (Gravier *et al.*, 2012) qui comportent 37 heures d'émissions de radio et de télévision (principalement des discours spontanés avec des locuteurs qui se chevauchent) ;

3) des données des campagnes d'évaluation ESTER 1 & ESTER 2 (Galliano *et al.*, 2005) qui contiennent 111 heures d'enregistrement audio transcrit. Ce sont principalement des programmes de radio français et africains (mélange de discours préparés et plus spontanés : parole du présentateur, interviews, reportages) ;

4) les données de la campagne d'évaluation REPERE (Kahn *et al.*, 2012) :
54 heures d'émissions transcrites de parole spontanée (des débats TV) et de la parole préparée (journaux télévisés).

Comme décrit dans le tableau 1, nos données contiennent de la parole non spontanée (NS) et de la parole spontanée (S). Les données d'entraînement (Train$_{SRAP}$) de notre système de transcription de la parole automatique sont sélectionnées à partir des données non spontanées qui correspondent essentiellement à des journaux télévisés. Les données utilisées pour la tâche de prédiction (*Train*$_{pred}$ et *Test*$_{pred}$) sont un mélange des deux styles de parole (S et NS). Il est important de mentionner que les émissions du corpus *Test*$_{Pred}$ n'existent pas dans le *Train*$_{Pred}$ et *vice versa*. En outre, des émissions plus difficiles (ayant des taux d'erreurs plus élevés) ont été sélectionnées pour *Test*$_{Pred}$. La distribution détaillée des taux d'erreurs sur notre corpus Test$_{pred}$ est donnée plus loin dans la figure 3. Dans le tableau 2, nos émissions ayant un style de parole spontanée ont systématiquement un taux d'erreurs plus élevé (de 28,74 % à 45,15 % selon l'émission) par rapport aux émissions ayant un style de parole non spontanée (de 12,06 % à 25,41 % selon l'émission). Cette division S et NS nous permettra de comparer nos systèmes de prédiction de performance sur différents types de documents contenant du discours NS et S.

2. http ://www.quaero.org

	Train_{SRAP}	Train_{Pred}	Test_{Pred}
NS	100 h 51	30 h 27	4 h 17
S	-	59 h 25	4 h 42
Durée	100 h 51	89 h 52	8 h 59
TEM	-	**22,29**	**31,20**

Tableau 1. *Distribution de nos corpus entre les styles de parole non spontanée (NS) et spontanée (S) – TEM = taux d'erreurs mots (WER)*

Source	Émission	Mots	TEM
	Non spontanées (NS)		
Quaero	Arte News (AN)	3 726	12,06
ESTER 2	Tvme (T)	10 706	18,44
Quaero	France Culture TEMPS (FCT)	10 091	20,92
Quaero	Fab histoire (FH)	10 022	22,76
ESTER 2	Africa1 (A1)	15 257	25,41
	Spontanées (S)		
Quaero	Ce soir ou jamais (CSOJ)	10 992	28,74
REPERE	Planete showbiz (PS)	15 946	36,74
REPERE	Culture et vous (CV)	16 026	39,79
ETAPE	La place du village (PV)	20 396	45,15

Tableau 2. *Performance sur le corpus Test_{Pred} en termes de TEM (taux d'erreurs de mots)*

3.2. *Métriques d'évaluation*

Nous avons utilisé la boîte à outils LNE-Tools (Galibert, 2013) afin d'évaluer la qualité de notre système de transcription et produire les rapports de performance en termes de taux d'erreurs de mots (TEM). La parole superposée et les tours de parole vides sont supprimés. Comme il est mentionné dans le tableau 1, nous avons obtenu 22,29 % de TEM sur le corpus Train_{pred} et 31,20 % de TEM sur le corpus Test_{pred}.

Afin d'évaluer la tâche de prédiction de performance, nous utilisons la métrique *Mean Absolute Error* (MAE) définie comme suit :

$$MAE = \frac{\sum_{i=1}^{N} |TEM_{Ref}^{i} - TEM_{Pred}^{i}|}{N} \quad [1]$$

avec N le nombre d'unités (tours de parole ou document complet).

Nous utilisons également le coefficient de corrélation de rang Kendall entre le score de référence et la sortie du système de prédiction au niveau des tours de parole. Plus le score Kendall est proche de 1, plus les performances prédites sont proches des vraies performances mesurées.

3.3. *Système de reconnaissance automatique de la parole*

Afin de produire des transcriptions automatiques pour le système de prédiction de performance, nous avons construit un système de transcription automatique de la parole fondé sur la boîte à outils KALDI (Povey *et al.*, 2011), en suivant la recette standard. Un système hybride HMM-DNN a été appris en utilisant le corpus Train_{SRAP} (100 heures de journaux diffusées par ESTER, REPERE, ETAPE et Quaero). Nous avons entraîné un modèle de langue 5-grammes à partir de plusieurs corpus français (3 milliards de mots au total : EUbookshop, TED2013, Wit3, GlobalVoices, Gigaword, Europarl-v7, MultiUN, OpenSubtitles2016, DGT, News Commentary, News WMT, *Le Monde*, Trames, Wikipédia) et les transcriptions de notre jeu de données $Train_{SRAP}$ en utilisant l'outil SRILM (Stolcke *et al.*, 2002). Pour le modèle de prononciation, nous avons utilisé la ressource lexicale BDLEX (De Calmès et Pérennou, 1998) ainsi que l'outil de conversion automatique de graphèmes à phonèmes LIA_Phon[3] afin de trouver les variantes de prononciation de notre vocabulaire (limité à 80 k mots).

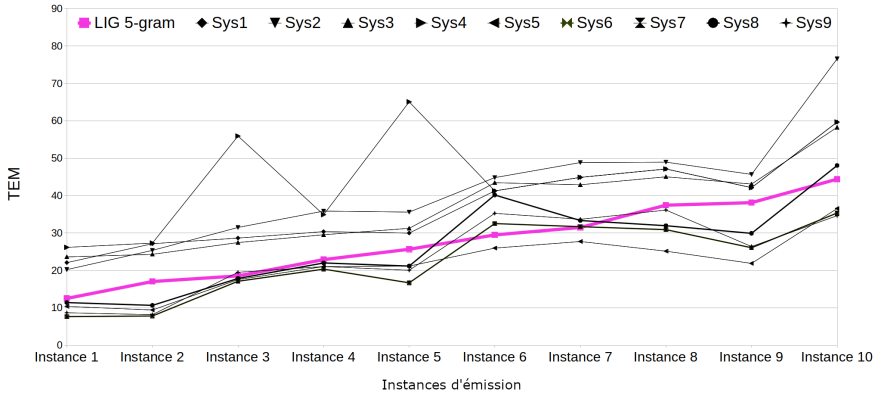


Figure 1. *Performance (TEM) de notre SRAP comparé aux performances d'autres systèmes sur des données identiques. Chaque instance en abscisse représente une instance d'un type d'émission.*

Dans la figure 1, nous comparons les performances de notre système LIG 5-gram (colorées en rose) à celles obtenues au cours des différentes campagnes d'évaluation (colorées en noir) sur les mêmes instances d'émissions (10 au total). Le système que nous avons développé est situé au milieu des systèmes proposés. Cela signifie que notre système produit des transcriptions correctes et que les performances sont corrélées avec celles des autres systèmes.

3. `http://lia.univ-avignon.fr/chercheurs/bechet/download/lia_phon.v1.2.jul06.tar.gz`

4. Prédiction de performance

Dans cette section, nous présentons une étude comparative entre une approche de prédiction fondée sur des caractéristiques explicites (système de base) et une nouvelle approche de prédiction fondée sur des caractéristiques entraînées en utilisant un réseau de neurones convolutif. Les transcriptions automatiques des corpus Train$_{pred}$ et Test$_{pred}$ ont été obtenues par notre système SRAP présenté dans la section 3.3. Les rapports de performance ont été générés par l'outil LNE-Tool (un taux d'erreurs de mots par tour de parole).

4.1. *Prédiction fondée sur des traits explicites (baseline)*

Afin d'avoir un système de prédiction de l'état de l'art (extraction des traits prédéfinis), nous avons adapté l'outil TranscRater (Jalalvand *et al.*, 2016) de l'anglais vers le français. Ce dernier s'appuie sur l'extraction de traits explicites (*engineered features*) pour prédire la performance de chaque entrée en termes de TEM. En exploitant des résultats empiriques antérieurs dans (Negri *et al.*, 2014 ; de Souza *et al.*, 2013 ; Jalalvand *et al.*, 2015b ; de Souza *et al.*, 2015), TranscRater exploite l'algorithme Extremely Randomized Trees (Geurts *et al.*, 2006) pour l'apprentissage du système. La sélection des traits est effectuée avec l'algorithme Randomized Lasso (Meinshausen et Bühlmann, 2010). Les principaux hyperparamètres du modèle sont optimisés à l'aide d'une grille de recherche avec une validation croisée sur l'ensemble des données d'apprentissage, afin de minimiser l'erreur absolue moyenne (MAE) entre les vrais TEM et les TEM prédits.

TranscRater est capable d'extraire 63 traits de quatre types :

– **9 traits morphosyntaxiques (POS)** : permet de capturer la plausibilité de la transcription d'un point de vue syntaxique en utilisant l'outil Treetagger (Schmid, 1995). Pour chaque mot compris dans un tour de parole transcrit, un score de prédiction d'étiquette POS est attribué au niveau du mot lui-même ainsi qu'au précédent et au suivant. Cette fenêtre glissante de 3 mots est utilisée pour calculer la valeur moyenne de l'ensemble du tour de parole transcrit. De plus, le vecteur de traits comporte également le nombre et le pourcentage de classes de *tokens* (nombres, noms, verbes, adjectifs et adverbes). Ces traits ont été testés dans diverses conditions (données propres ou bruitées, microphones simples ou multiples) (Jalalvand *et al.*, 2015a ; Jalalvand *et al.*, 2015c).

– **3 traits issus du modèle de langue (LM)** : permet de capturer la plausibilité de la transcription selon un modèle n-gramme. Ils comprennent la moyenne des probabilités des mots, la somme des log-probabilités et le score de perplexité pour chaque transcription. Un modèle 5-grammes est entraîné en utilisant l'outil SRILM (Stolcke *et al.*, 2002) sur l'ensemble des corpus textes de 3 milliards de mots mentionné dans la section 3.3 ;

– **7 traits lexicaux (LEX)** : les traits sont extraits à partir du lexique de notre système de transcription : un vecteur de traits contenant la fréquence des catégories de phonèmes liées à la prononciation de chaque mot ;

– **44 traits acoustiques (SIG)** : ils capturent des informations sur le signal d'entrée (conditions générales d'enregistrement, accents spécifiques au locuteur). Pour l'extraction des traits, TranscRater calcule 13 paramètres de type MFCC (en utilisant openSMILE (Eyben *et al.*, 2010)), leurs dérivées, accélération et log-énergie, fréquence fondamentale (F0), probabilité de voisement, contours d'intensité et le *pitch* pour chaque trame de parole. Pour l'ensemble du signal d'entrée, le vecteur de traits SIG est obtenu en calculant la moyenne des valeurs de chaque trame.

4.2. *Prédiction par les réseaux neuronaux convolutifs (CNN)*

Afin de prédire le TEM, nous proposons une nouvelle approche de régression supervisée fondée sur des réseaux de neurones convolutifs. Notre réseau prend en entrée des données textuelles et/ou des données acoustiques (signal brut, des MFCC ou des spectrogrammes). Suivant notre protocole expérimental, le système de RAP est considéré comme une boîte noire, et seuls les signaux et/ou les transcriptions automatiques sont fournis pour créer et évaluer des systèmes de prédiction. Nous avons construit notre modèle en utilisant à la fois Keras (Chollet *et al.*, 2015) et Tensorflow[4].

Pour l'entrée textuelle, nous proposons une architecture inspirée de Kim (2014) (verte dans figure 2). L'entrée est un tour de parole complété à N mots (N est défini comme la longueur de la plus longue phrase dans notre corpus complet) présenté sous forme d'une matrice EMBED de taille N $\times$ M (M = la dimension des représentations vectorielles de mots). Ainsi, chaque ligne de la matrice EMBED correspond à une représentation vectorielle (appelée aussi *embeddings*) d'un mot. Ces *embeddings* ont été initialisés à l'aide d'un modèle pré-entraîné sur l'ensemble des corpus textuels (3 milliards de mots mentionnés dans la section 3.3) en utilisant l'outil Word2Vec (Mikolov *et al.*, 2013) puis, mis à jour automatiquement au moment de l'apprentissage du réseau.

Chaque opération de convolution implique un filtre w qui est appliqué à un segment de h mots pour produire une nouvelle caractéristique. Par exemple, la caractéristique c_i est générée à partir des mots $x_{i:i+h-1}$ comme :

$$c_i = f(w \cdot x_{i:i+h-1} + b) \quad [2]$$

où b est un terme de biais et f est une fonction non linéaire. Ce filtre est appliqué à chaque segment du mot dans le tour de parole pour produire un vecteur (*feature map*) $c = [c_1, c_2 ... c_{n-h+1}]$. Ensuite, une opération de type *max-pooling* (Collobert *et al.*, 2011) prend les 4 plus grandes valeurs de c, qui sont ensuite moyennées. L'ensemble des filtres W fournit une entrée de taille fixe (nombre de filtres de W * nombre de

4. `https://www.tensorflow.org`

régions de convolution) aux deux couches cachées entièrement connectées (256 et 128 unités) suivies respectivement d'une régularisation de type $dropout$ (0,2 et 0,6) avant la prédiction de la performance (TEM).

Pour l'entrée du signal, nous utilisons la meilleure architecture proposée dans (Dai *et al.*, 2017) (colorée en rouge dans la figure 2). Il s'agit d'un CNN profond avec 17 couches convolution + *max-pooling* suivies d'une opération d'agrégation (*global average pooling)* et de trois couches cachées complètement connectées (512, 256 et 128 unités). Nous avons ajouté une régularisation (*dropout*) de 0,2 entre les deux dernières couches (256 et 128). Nous proposons plusieurs méthodes pour encoder le signal avec le CNN en utilisant Librosa (McFee *et al.*, 2015) : les échantillons du signal brut (RAW - SIG), le spectrogramme (MEL-SPEC) ou des coefficients MFCC.

Afin de prédire un taux d'erreurs (TEM) à l'aide des réseaux CNN, nous proposons deux approches différentes :

– **CNN**$_{Softmax}$: nous utilisons les probabilités Softmax et un vecteur fixe externe nommé TEM$_{Vector}$ pour calculer le TEM prédit (TEM$_{Pred}$). TEM$_{Vector}$ et les probabilités Softmax doivent avoir la même dimension. TEM$_{Pred}$ est alors défini comme suit :

$$TEM_{Pred} = \sum_{C=1}^{NC} P_{Softmax}(C) * TEM_{Vector}(C) \quad [3]$$

NC est la dimension du vecteur TEM$_{Vector}$. Dans nos expériences, NC est égale à 6 et TEM$_{Vector}$ = [0 %, 25 %, 50 %, 75 %, 100 %, 150 %] ;

– **CNN**$_{ReLU}$: nous appliquons la fonction ReLU (la taille de sortie est égale à 1) à la dernière couche cachée du réseau. Cette fonction permettra d'estimer directement le TEM en retournant une valeur de type réel entre 0 et $+\infty$.

Pour l'utilisation jointe des données textuelles et acoustiques, nous fusionnons les deux dernières couches cachées de CNN EMBED et CNN RAW - SIG (ou MEL-SPEC ou MFCC) en les concaténant et en les faisant passer à une nouvelle couche cachée (de taille 128) avant la prédiction de TEM avec le CNN$_{Softmax}$ ou le CNN$_{ReLU}$ (représentées par des lignes en pointillé dans la figure 2). Nous entraînons par la suite le réseau de la même manière.

Contrairement aux traits de l'approche de base (extraction qui nécessite d'avoir défini les traits au préalable, on parle dans ce cas d'*engineered features*), les traits CNN textuels sont extraits et entraînés à partir des représentations vectorielles des mots (on parle alors de *learnt features*). Ces traits sont appris par le réseau neuronal jusqu'à ce que le comportement désiré soit obtenu.

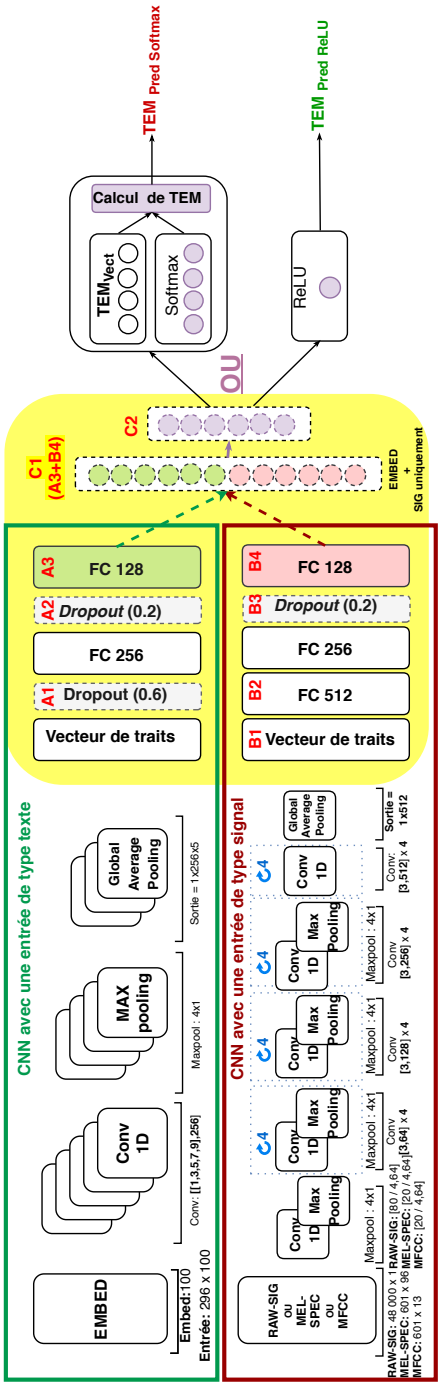


Figure 2. *Architecture de nos CNN à partir d'entrées texte (vert) et signal (rouge). Les couches avec des lignes en pointillé correspondent à l'utilisation conjointe texte + signal*

4.3. *Expériences et résultats*

Dans cette section, nous comparons les deux approches de prédiction de performance des SRAP : prédiction fondée sur des caractéristiques explicites et prédiction fondée sur des caractéristiques entraînées en utilisant les CNN. La prédiction par l'outil TranscRater s'appuie sur les traits issus de la sortie du SRAP et du signal (POS, LEX, LM et SIG), tandis que le CNN est fondé sur la sortie du SRAP et le signal brut. Pour le CNN, nous sélectionnons aléatoirement 10 % des données Train$_{Pred}$ comme corpus de développement DEV. Le reste est considéré comme un corpus d'apprentissage du réseau (TRAIN). Dix modèles de prédiction sont entraînés selon 10 sélections aléatoires de la partition TRAIN et DEV.

L'entraînement est effectué à l'aide de l'algorithme Adadelta (Zeiler, 2012) sur des mini-batchs de taille 32 avec 50 époques d'apprentissage.[5] La métrique MAE est utilisée à la fois comme fonction de perte (coût) et comme mesure d'évaluation. Après la phase d'apprentissage, nous prenons le meilleur modèle (parmi les 10 sélections aléatoires de la partition TRAIN et DEV) obtenu en termes de MAE sur le corpus DEV et nous l'évaluons sur le corpus Test$_{Pred}$.

Nous étudions plusieurs entrées pour le CNN :

1) entrées textuelles (transcription automatique) uniquement (**EMBED**) : l'entrée du réseau est une matrice de dimension 296 $\times$ 100 (296 est la longueur de l'hypothèse SRAP la plus longue dans notre corpus ; 100 est la dimension des *embeddings* de mots pré-entraînés sur le grand corpus de texte de 3,3 milliards de mots). Nous utilisons des tailles différentes de fenêtres de filtre *h* de [1, 3, 5, 7, 9] avec des filtres de taille 256 ;

2) signal brut uniquement (**RAW - SIG**) : les modèles sont entraînés sur des tours de parole de 6 secondes et échantillonnés à 8 kHz seulement (pour éviter les problèmes de surcharge de mémoire au cours de l'apprentissage du CNN). Les tours de parole courts ($< 6s$) sont complétés par des zéros (silence). Notre entrée est un vecteur de dimension 48 000 $\times$ 1. Les paramètres des filtres sont détaillés dans la figure 2 ;

3) spectrogramme seulement (**MEL-SPEC**) : nous utilisons la même configuration que pour le signal brut ; nous avons des vecteurs en entrée de dimension 96 (chaque dimension correspond à une plage de fréquence particulière) extraits toutes les 10 ms (la fenêtre d'analyse est de 25 ms). Notre entrée a donc une dimension 601 $\times$ 96,[6] ;

4) paramètres **MFCC** seulement : nous calculons 13 MFCC[7] toutes les 10 ms pour fournir au réseau CNN une entrée de dimension 601 $\times$ 13 ;

5) entrées conjointes (texte et signal) (EMBED + RAW - SIG ou EMBED + MEL - SPEC ou EMBED + MFCC) : dans ce cas, nous concaténons les dernières couches

5. L'algorithme d'optimisation et le nombre d'époques sont définis suivant les performances obtenues sur le corpus de DEV.

6. Les paramètres détaillés des filtres sont représentés dans la figure 2

7. Par souci de comparaison avec Transrater, les dérivées premières et secondes ne sont pas prises en compte.

cachées des réseaux CNN texte et signal (lignes en pointillé dans la figure 2).

4.3.1. *Résultats*

Les lignes TranscRater du tableau 3 présentent les résultats obtenus avec le système de base fondé sur la régression. Nous pouvons observer que la meilleure performance est obtenue avec les caractéristiques textuelles POS + LEX + LM (MAE de 22,01 %) alors que l'ajout du SIG n'améliore pas le modèle (MAE de 21,99 %). Cette impossibilité à intégrer correctement les caractéristiques issues du signal, dans les modèles de TranscRater , a également été observée par Jalalvand *et al.* (2016).

Modèle	Input	MAE	Kendall
Caractéristiques textuelles (TXT)			
TranscRater	POS + LEX + LM	22,01	**44,16**
CNN$_{Softmax}$	EMBED	**21,48**	38,91
CNN$_{ReLU}$	EMBED	22,30	38,13
Caractéristiques acoustiques (SIG)			
TranscRater	SIG	25,86	23,36
CNN$_{Softmax}$	RAW - SIG	25,97	23,61
CNN$_{ReLU}$	RAW - SIG	26,90	21,26
CNN$_{Softmax}$	MEL - SPEC	29,11	19,76
CNN$_{ReLU}$	MEL - SPEC	26,07	24,29
CNN$_{Softmax}$	MFCC	**25,52**	**26,63**
CNN$_{ReLU}$	MFCC	26,17	25,41
Caractéristiques textuelles et acoustiques (TXT + SIG)			
TranscRater	POS + LEX + LM + SIG	21,99	45,82
CNN$_{Softmax}$	EMBED + RAW - SIG	**19,24**	**46,83**
CNN$_{ReLU}$	EMBED + RAW - SIG	20,56	45,01
CNN$_{Softmax}$	EMBED + MEL-SPEC	20,93	40,96
CNN$_{ReLU}$	EMBED + MEL-SPEC	20,93	44,38
CNN$_{Softmax}$	EMBED + MFCC	19,97	44,71
CNN$_{ReLU}$	EMBED + MFCC	20,32	45,52

Tableau 3. *TranscRater vs CNN*$_{Softmax}$ *vs CNN*$_{ReLU}$ *évalués au niveau de la phrase avec une métrique MAE ou Kendall sur le corpus Test*$_{pred}$

En utilisant des caractéristiques textuelles uniquement, nous constatons que CNN$_{Softmax}$ et CNN$_{ReLU}$ ont des performances équivalentes (meilleurs en termes de MAE mais moins performantes en termes de Kendall) par rapport au modèle de TranscRater . CNN$_{Softmax}$ montre une meilleure performance que CNN$_{ReLU}$ en termes de MAE et de coefficient de corrélation.

Toutefois, il faut noter aussi que la tâche de prédiction de performance est difficile en s'appuyant essentiellement sur les caractéristiques acoustiques (MAE supé-

rieur à 25 %). Cependant, parmi les différentes entrées du signal testées, de simples MFCC conduisent à une meilleure performance en termes de MAE et Kendall. Bien que l'utilisation conjointe de caractéristiques textuelles et acoustiques n'ait pas donné des bons résultats pour la prédiction par TranscRater , elle mène à de meilleures performances en utilisant les CNN. La meilleure performance est obtenue avec le système CNN$_{Softmax}$ (EMBED + RAW - SIG) [8] qui dépasse l'approche de régression (le MAE est réduit de 21,99 % à 19,24 %, et la corrélation entre les vrais TEM et les TEM prédits est améliorée de 45,82 % à 46,83 % en termes de Kendall). Un *test de Wilcoxon* [9] permet de confirmer que la différence entre les TEM prédits par ces deux systèmes est significative ($p < 0,001$).

4.3.2. *Analyse des taux d'erreurs de mots (TEM) prédits*

Le tableau 4 présente les TEM prédits sur le corpus TEST en utilisant les deux approches de prédiction (TranscRater et CNN) pour les différents styles de parole (spontanée et non spontanée). Les performances montrent que notre approche (à − 3,83 % du TEM références) est meilleure que l'approche de régression (− 5,38 %) sur l'ensemble du corpus. Les performances montrent que le système CNN a bien prédit le TEM sur la parole non spontanée et spontanée. Le TEM$_{Pred}$ est à − 2,54 % sur la parole non spontanée et à − 4,84 % sur la parole spontanée. En revanche, la méthode de régression n'arrive pas à bien prédire la performance sur la parole spontanée (− 10,11 %).

	NS	S	NS + S
TEM$_{REF}$	21,47	38,83	31,20
TEM$_{Pred}$ TranscRater	**22,08**	28,72	25,82
TEM$_{Pred}$ CNN$_{Softmax}$	18,93	**33,99**	**27,37**
#Tours Parole	3,1 k	3,7 k	6,8 k
#Mots$_{REF}$	49,8 k	63,3 k	113,1 k

Tableau 4. *TranscRater (POS + LEX + LM + SIG) vs CNN$_{Softmax}$ (EMBED + RAW - SIG) des TEM prédits (moyennés sur toutes les phrases) par type de parole (NS ou S) sur le corpus Test$_{pred}$*

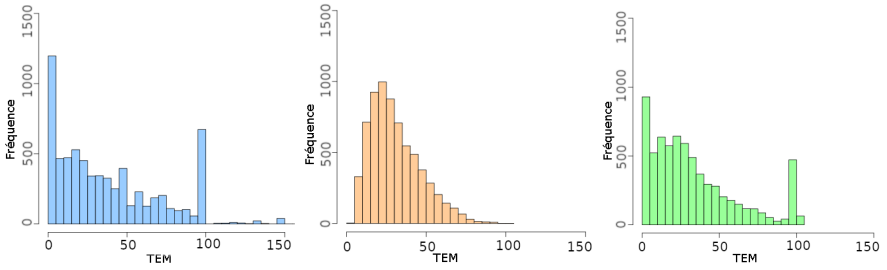
La figure 3 présente l'analyse de prédiction de TEM au niveau des tours de parole.[10] Elle montre la distribution des tours de parole en fonction de leur TEM réel ou prédit. Il est clair que la prédiction CNN permet d'approximer la vraie distribution de TEM sur le corpus Test$_{pred}$. La distribution produite par TranscRater ressemble à une

8. Les MAE obtenus sur les 10 modèles atteignent entre 15,24 % et 15,96 % sur les corpus de développement, tandis que les MAE obtenus sur le corpus Test$_{Pred}$ sont entre 19,24 % et 20,70 %

9. `http://www.r-tutor.com/elementary-statistics/non-parametric-methods/wilcoxon-signed-rank-test`

10. Les sorties des modèles sont disponibles sur `http://www.lne.fr/LNE-LIG-WER-Prediction-Corpus`

distribution gaussienne autour de la moyenne TEM observée sur les données d'apprentissage. Il est également intéressant de relever que les deux pics de TEM = 0 % et TEM = 100 % sont prédits correctement par notre système CNN.



(a) Référence (b) TranscRater (c) CNN

Figure 3. *Distribution des tours de parole en fonction de leurs TEM : (a) référence (b) prédit par le meilleur système TranscRater, (c) prédit par le meilleur système CNN*

Dans la figure 4, nous comparons les meilleurs systèmes de prédiction CNN et TR en termes de MAE sur le corpus Test_{pred} au niveau du type d'émission afin de comprendre l'effet du style de parole sur la tâche de prédiction de performance. Comme décrit, nous avons classé les émissions de nos corpus en deux groupes : parole non spontanée (NS) et parole spontanée (S). Les performances obtenues confirment que la performance sur la parole spontanée est plus difficile à prédire que sur la parole non spontanée. Dans la partie spontanée, nous remarquons que l'écart entre la courbe CNN et la courbe TR est plus large que pour la parole non spontanée. Cela signifie que le système CNN est capable de prédire un TEM élevé, alors que le système TR prédit une performance autour du TEM moyen observé sur les données d'entraînement Train_{pred}.

5. Évaluation des représentations apprises

5.1. *Méthodologie*

Dans cette section, nous essayons de comprendre ce que notre meilleur système de prédiction de performance (EMBED + RAW - SIG) a appris. Nous analysons les représentations textuelles et acoustiques obtenues par notre architecture. Nous nous inspirons de travaux de Belinkov et Glass (2017) : le modèle pré-entraîné (EMBED + RAW - SIG) est utilisé pour générer des représentations au niveau des tours de parole. Nous nous intéressons à l'analyse des représentations qui correspondent à différentes couches supérieures de notre réseau (colorées en jaune dans la figure 2). Ces représentations sont utilisées par la suite pour entraîner un classifieur peu profond et résoudre des tâches de classification annexes telles que :

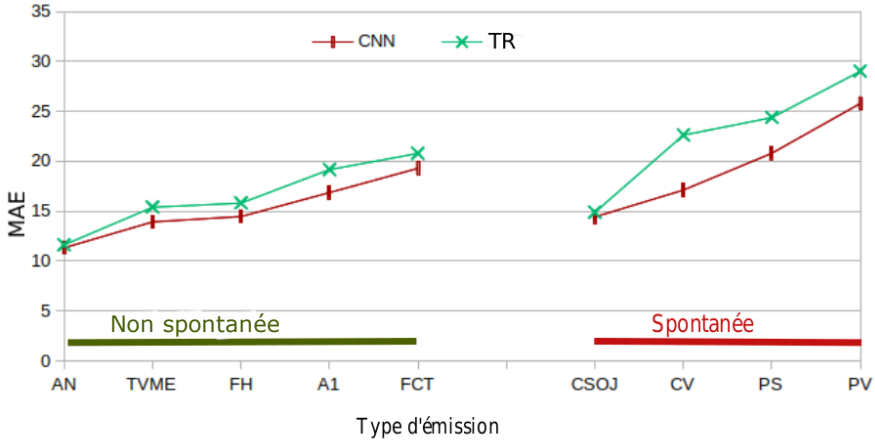


Figure 4. *Évaluation des systèmes de prédiction sur le corpus $Test_{pred}$ en termes de MAE au niveau du type d'émission*

– **STYLE :** classer les tours de parole entre les styles de parole (S et NS) (voir le tableau 5) ;

– **ACCENT :** classer les tours de parole entre locuteurs natif et non natif (comme il est indiqué dans le tableau 5), nous avons utilisé les annotations des locuteurs fournies avec nos données afin d'étiqueter nos tours de parole entre natifs et non natifs ;

– **ÉMISSION :** classer les tours de parole suivant les émissions. Comme cela est décrit dans le tableau 6, chaque tour de parole de notre corpus est étiqueté avec le nom de l'émission.

Les performances de ces classifieurs peu profonds nous permettront de savoir quelles informations (style, accent, émission) sont le mieux capturées par quelles couches du réseau ; c'est-à-dire ce que modélise un réseau CNN qui prédit les performances d'un SRAP.

Comme analyse visuelle, nous projetons également un exemple des représentations dans un espace à deux dimensions à l'aide de l'algorithme *t-sne* (t-distributed stochastic neighbor embedding).[11]

5.2. *Classifieur peu profond pour l'analyse*

Nous avons construit trois classifieurs peu profonds (ÉMISSION, STYLE, ACCENT) avec une architecture similaire. Le classifieur est un réseau neuronal supervisé

11. `https://lvdmaaten.github.io/tsne/code/tsne_python.zip`

avec une seule couche cachée (la taille de la couche cachée est fixée à 128) suivie d'un *dropout* (taux de 0,5) et d'une non-linéarité *ReLU*. Enfin, une couche Softmax est utilisée afin de convertir la sortie du réseau en une catégorie prédite. Nous avons choisi un classifieur simple et peu profond car nous nous intéressons à l'évaluation de la qualité des représentations apprises par notre modèle de prédiction SRAP, plutôt qu'à l'optimisation des tâches de classification secondaires. La taille de l'entrée du réseau dépend de la couche à analyser (voir figure 2).

L'apprentissage est effectué en utilisant l'algorithme Adam (Kingma et Ba, 2014) (en utilisant les paramètres par défaut) sur des mini-batchs de taille 16. La fonction de coût est l'entropie croisée. Les modèles sont entraînés avec 30 époques. Après l'apprentissage, nous conservons le modèle ayant les meilleures performances sur l'ensemble DEV et nous l'évaluons sur le corpus TEST (voir section suivante pour les détails sur DEV et TEST). Les sorties du classifieur sont évaluées en termes de taux de bonne classification (*accuracy*).

5.3. *Données*

Nous avons utilisé les mêmes données que celles proposées dans la section 3.1. Nous récupérons tout d'abord les corpus d'apprentissage (TRAIN) et de développement (DEV) du meilleur modèle obtenu (EMBED + RAW - SIG), tout en gardant le même corpus de TEST (Test$_{Pred}$), sachant que les émissions du corpus Test$_{Pred}$ n'existent ni dans le corpus TRAIN ni dans le corpus DEV.

Catégorie	**TRAIN**	**DEV**	**TEST**
Non spontanée	54 250	6 101	**3 109**
Spontanée	**13 277**	**1 403**	3 728
Native	44 487	4 945	5 298
Non native	**23 040**	**2 559**	**1 539**

Tableau 5. *Distribution de nos tours de parole entre des styles non spontanés et spontanés, accents natifs et non natifs*

Émission	**TRAIN**	**DEV**	**TEST**
FINTER-DEBATE	7 632	833	-
FRANCE3-DEBATE	928	77	-
LCP-PileEtFace	**4 487**	525	-
RFI	25 565	2 831	-
RTM	24 198	2 745	-
TELSONNE	4 717	**493**	-
Total	67 527	7 504	-

Tableau 6. *Nombre des tours de parole pour chaque émission*

Les tableaux 5 et 6 décrivent l'ensemble des données disponibles en termes de tours de parole pour chaque tâche de classification. Nous constatons clairement que les données sont déséquilibrées pour les trois catégories (STYLE, ACCENT, ÉMISSION). Étant donné que nous nous intéressons à évaluer le pouvoir discriminant de nos représentations apprises pour ces trois tâches, nous avons extrait une version équilibrée de nos données TRAIN, DEV, TEST en filtrant les étiquettes surreprésentées (le nombre final de tours de parole conservés correspond aux nombres en gras dans les tableaux 5 et 6). Le corpus TEST ne contient aucun type d'émission présent dans le tableau 6, car selon notre protocole expérimental, les émissions du corpus TEST (voir tableau 2) n'existent pas dans les corpus TRAIN et DEV et *vice versa*.

5.4. *Résultats*

Pour chaque tâche de classification, nous avons construit un classifieur peu profond en utilisant les représentations cachées des caractéristiques EMBED (texte), RAW - SIG (signal) et EMBED + RAW - SIG en entrée. Le tableau 7 présente les résultats expérimentaux obtenus sur les corpus DEV et TEST séparés par deux barres verticales (||). Les performances des systèmes de classification sont toutes supérieures à un taux de bonne classification correspondant à une décision aléatoire (> 50 % pour les tâches STYLE et ACCENT et > 20 % pour la tâche ÉMISSION). Cela montre que l'apprentissage d'un système de prédiction de TEM profond produit des représentations (au niveau des couches) qui contiennent une quantité significative d'informations sur le style de parole, l'accent du locuteur ainsi que sur le type d'émission. La prédiction du style des tours de parole (spontanée et non spontanée) est légèrement plus facile que la prédiction de l'accent (natifs et non-natifs), en particulier à partir de l'entrée de type texte (EMBED). Cela pourrait être lié à la durée courte (< 6 s) des tours de parole, étant donné que l'identification de l'accent a probablement besoin de séquences plus longues. Nous observons également que l'utilisation du texte et de la parole améliore les représentations apprises pour la tâche STYLE alors que cela est moins clair pour la tâche ACCENT (étant donné que l'amélioration observée sur DEV n'est pas confirmée sur TEST).

Enfin, l'entrée textuelle est significativement meilleure que l'entrée acoustique pour toutes les tâches de classification, alors que nous anticipions de meilleures performances sur l'entrée acoustique pour la tâche ÉMISSION (le signal transmet des informations sur les caractéristiques acoustiques d'un programme diffusé). Parmi les représentations analysées, les sorties des CNN (A1, B1) conduisent aux meilleurs résultats de classification, ceci est cohérent avec les résultats de la littérature qui présentent les convolutions comme de bons extracteurs de traits. En utilisant les couches supérieures (entièrement connectées), nous remarquons que la performance se dégrade. Cela signifie que l'information sur le style de parole, l'accent du locuteur ou l'émission et plutôt capturée dans les couches moins hautes de notre architecture neuronale de prédiction de performance de SRAP.

Couche	Dim.	ÉMISSION	STYLE	ACCENT
		EMBED		
A1	1 280	**57,12**||-	**80,72**||68,99	**70,75**||66,54
A2	256	54,89||-	80,01||**69,56**	69,30||69,43
A3	128	51,04||-	79,23||68,27	68,25||**70,89**
		RAW - SIG		
B1	512	**42,35**||-	**72,92**||**58,64**	64,60||**55,85**
B2	512	41,22||-	72,20||58,41	64,44||54,84
B3	256	41,22||-	72,38||58,44	64,50||54,65
B4	128	40,77||-	72,38||58,52	**64,74**||54,87
		EMBED + RAW - SIG		
C1 (A3+B4)	256	**57,04**||-	**81,29**||70,36	**71,41**||**65,98**
C2	128	53,06||-	79,62||**70,55**	70,01||65,20
Aléatoire	-	**20,00**	**50,00**	**50,00**

Tableau 7. *Performances des systèmes de classification émission (sur le corpus DEV uniquement), style et accent en termes de taux de bonne classification en utilisant les représentions apprises durant l'apprentissage de notre système de prédiction*

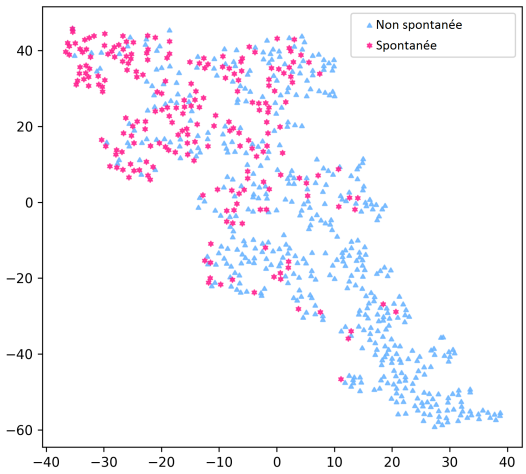
Dans la figure 5, nous visualisons un exemple de représentations des tours de parole de la couche C2 (EMBED + RAW - SIG) en utilisant t-SNE. Pour une durée fixe[12] de 4 à 5 s (716 tours de parole) et de 5 à 6 s (489 tours de parole), les tours de parole non spontanée sont colorés en bleu tandis que les tours de parole spontanée sont en rose. La couche C2 produit des *clusters* qui montrent que les tours de parole spontanée se trouvent dans la partie supérieure gauche de l'espace 2D. Cela suggère que la représentation cachée C2 véhicule une information (signal faible) sur le style de parole.

Enfin, la figure 6 présente la matrice de confusion produite à l'aide de la couche C2 (EMBED + RAW - SIG). Les classifieurs ont très bien prédit la catégorie *TELSONNE* (taux de bonne classification de 82 %), qui contient de nombreux appels téléphoniques des auditeurs de la radio. Cette émission est assez différente des quatre autres émissions de DEV (débats et actualités).

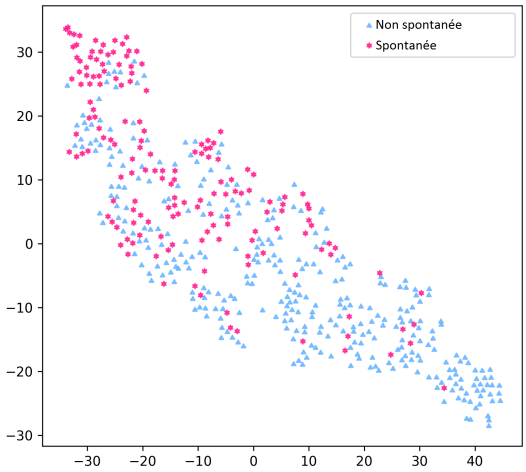
6. Apprentissage multitâche

Dans la section précédente, nous avons montré que les couches cachées de notre système de prédiction capturaient une information sur le style de la parole, l'accent et le type d'émission. Cela suggère que ces trois types d'informations pourraient être utiles pour structurer l'apprentissage des modèles neuronaux de prédiction de perfor-

12. D'après nos expériences, les représentations des tours de parole très courts (ayant une durée inférieure à 2 s) capturent plus difficilement l'information sur le style de parole.



(a) de 4 à 5 secondes



(b) de 5 à 6 secondes

Figure 5. *Visualisation des représentations des tours de parole de la couche C2 pour les différents styles de parole (spontanée, non spontanée) : (a) des tours de parole ayant une durée de 4 à 5 s et (b) de 5 à 6 s*

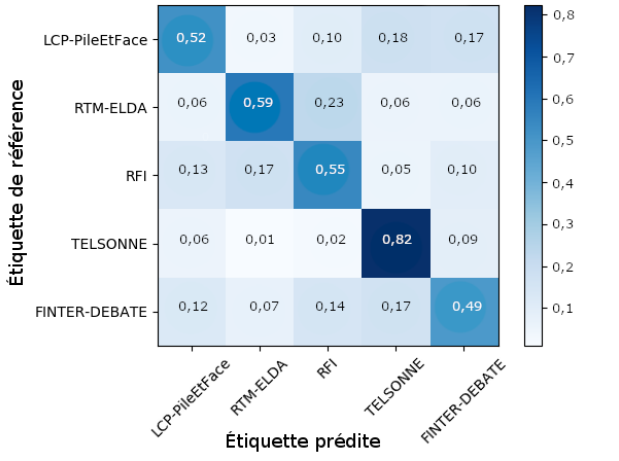


Figure 6. *Matrice de confusion de la classification ÉMISSION en utilisant les représentations de la couche C2 (EMBED + RAW - SIG) comme entrée - évaluée sur le corpus DEV*

mance. Dans cette section, nous examinons l'impact de la connaissance de ces étiquettes (style, accent, émission) au moment de l'apprentissage sur les performances des systèmes de prédiction. Pour cela, nous effectuons un apprentissage multitâche en fournissant des informations supplémentaires sur le type d'émission, le style de parole ainsi que l'accent du locuteur pendant l'apprentissage. L'architecture du modèle multitâche est similaire au modèle de prédiction de TEM (monotâche) présenté dans la figure 2 en ajoutant des sorties supplémentaires : une fonction Softmax est ajoutée pour chaque nouvelle tâche de classification après la dernière couche entièrement connectée (C2). La dimension de sortie dépend essentiellement de la tâche visée : 6 pour les tâches ÉMISSION et 2 pour les tâches STYLE et ACCENT.

Nous utilisons la totalité des données (non équilibrées) décrites dans les tableaux 5 et 6. L'entraînement du modèle multitâche utilise $Adadelta$. Les modèles sont appris pendant 50 époques avec une taille de mini-batch de 32. La métrique MAE est utilisée comme fonction de coût pour la tâche de prédiction, tandis que l'entropie croisée est utilisée pour les tâches de classification secondaire. Nous définissons aussi une fonction de coût composite dans le cas de l'apprentissage multitâche : nous attribuons une pondération de 1 pour le coût MAE (tâche principale) et une pondération plus petite de 0,3 pour le ou les coûts d'entropie croisée (tâche de classification secondaire).

Après la phase d'apprentissage, nous prenons le modèle qui donne le meilleur MAE sur le corpus DEV et nous l'évaluons sur le corpus TEST. Nous expérimentons plusieurs modèles qui traitent simultanément les 1, 2, 3 et 4 tâches. Les modèles sont

évalués avec une métrique spécifique pour chaque tâche : MAE et Kendall[13] pour la tâche de prédiction TEM et le taux de bonne classification (*accuracy*) pour les tâches de classification.

Les tableaux 8 et 9 résument les résultats expérimentaux sur les corpus DEV et TEST séparés par deux barres verticales (||). Nous avons considéré le modèle monotâche décrit dans la section 4.2 comme un système de référence.

Nous rappelons que nous avons évalué la tâche de classification ÉMISSION uniquement sur l'ensemble DEV (les émissions du corpus TEST n'existent pas dans notre TRAIN).

Tout d'abord, nous constatons que la performance des tâches de classification dans les scénarios multitâches est très bonne : nous sommes capables de former des systèmes efficaces de prédiction de performance SRAP qui annotent simultanément les tours de parole analysés en fonction de leur style de parole, de leur accent et de l'origine du programme de diffusion. De tels systèmes multitâches pourraient être utilisés comme outils de diagnostic pour analyser et prédire les TEM sur de grandes collections acoustiques.

De plus, nos meilleurs systèmes multitâches montrent une meilleure performance (MAE, Kendall) par rapport au système de base. Cela signifie que le fait de donner implicitement les informations sur le style, l'accent et le type d'émission peut être utile pour structurer l'apprentissage du système de prédiction.

Par exemple, pour les systèmes à deux tâches, le meilleur modèle est obtenu sur les tâches TEM + ÉMISSION avec une différence respective de + 0,41 % et + 2,25 % en termes de MAE et Kendall (sur le corpus DEV) par rapport au système de base sur la tâche de prédiction TEM.

Il faut cependant noter que l'impact de l'apprentissage multitâche sur la tâche principale (prédiction de la performance) est limité : de légères améliorations sur le corpus TEST sont observées en termes de MAE et Kendall. Néanmoins, les systèmes appris semblent complémentaires étant donné que leur combinaison (moyennage, sur l'ensemble des systèmes multitâches, du TEM prédit au niveau des tours de parole) conduit à une amélioration significative des performances (voir dernière ligne du tableau 8 pour MAE et Kendall).

13. Corrélation entre les vraies valeurs TEM (références) et les valeurs TEM prédites.

Tâche de prédiction de performance		
Modèles	**MAE**	**Kendall**
Baseline : monotâche		
TEM	15,24||19,24	45,00||46,83
2 tâches		
TEM ÉMISSION	**14,83**||19,15	**47,25**||47,05
TEM STYLE	15,07||19,66	45,92||45,49
TEM ACCENT	15,05||19,60	46,17||45,60
3 tâches		
TEM STYLE ACCENT	15,12||20,23	45,75||44,09
TEM ÉMISSION ACCENT	14,94||19,76	46,19||43,61
TEM ÉMISSION STYLE	14,90||**19,14**	45,87||**47,28**
4 tâches		
TEM ÉMISSION STYLE ACCENT	15,15||19,64	45,59||45,42
COMBINAISON TOUTES SORTIES	**14,50** ||**18,87**	**48,16**||**48,63**

Tableau 8. *Évaluation de la prédiction de performance du SRAP avec des modèles multitâches ($DEV||TEST$) en termes de MAE et Kendall*

Tâche de classification			
Modèles	**ÉMISSION**	**STYLE**	**ACCENT**
2 tâches			
TEM ÉMISSION	**99,29**||-	-	-
TEM STYLE	-	99,01||65,24	-
TEM ACCENT -	-	91,72|| 75,30	
3 tâches			
TEM STYLE ACCENT	-	98,63||69,07	88,99|| **77,46**
TEM ÉMISSION ACCENT	98,38||-	-	89,87||71,44
TEM ÉMISSION STYLE	99,12||-	**99,47**||**81,98**	-
4 tâches			
TEM ÉMISSION STYLE ACCENT	99,04||-	99,29||81,55	**91,92**||73,60

Tableau 9. *Évaluation des tâches de classification secondaires des modèles multitâches ($DEV||TEST$) en termes de taux de bonne classification*

7. Conclusion et perspectives

Dans ce travail, nous avons abordé la tâche de prédiction de performance des systèmes de transcription automatique de la parole. Dans un premier temps, nous avons proposé un corpus hétérogène en français spécifique pour cette tâche. Nous avons proposé par la suite de comparer deux différentes approches de prédiction de perfor-

mance : une approche fondée sur des traits prédéfinis (*engineered features*) en utilisant l'outil TranscRater et notre nouvelle approche fondée sur des traits estimés au cours de l'apprentissage d'un système neuronal de type CNN (*learnt features*). Nos expérimentations montrent que l'approche de prédiction par les CNN est meilleure que l'approche de prédiction de base (par TranscRater) en termes de scores MAE et Kendall. Plus précisément, l'utilisation conjointe en entrée des textes et signaux ne donne pas de résultats positifs pour les systèmes TranscRater , tandis qu'elle permet de meilleures performances en utilisant des CNN. Nous montrons également que les CNN prédisent correctement la distribution des taux d'erreurs de mots (TEM) sur une collection d'enregistrements, contrairement à TranscRater qui prédit une distribution proche d'une distribution gaussienne autour du TEM moyen observé dans le corpus d'apprentissage.

Dans un second temps, nous avons essayé de comprendre ce qu'apprend le système CNN en analysant les représentations intermédiaires produites par notre meilleur système de prédiction (CNN$_{Softmax}$ EMBED + RAW - SIG). Afin de comprendre quelles sont les informations capturées par le modèle au cours de l'entraînement, nous avons suivi une méthode d'analyse inspirée d'un article récent de Belinkov et Glass (2017) publié l'an dernier à la conférence NIPS. L'idée est d'utiliser les représentations apprises pour des tâches de classification annexes (ou de les visualiser). Nos expérimentations montrent que notre modèle capture des informations sur le style de parole, l'accent du locuteur et le type d'émission durant l'apprentissage du système. Enfin, nous avons étudié le potentiel d'un apprentissage structuré consistant à donner implicitement ces trois informations au moment de l'entraînement du système de prédiction via un apprentissage multitâche. Les performances obtenues montrent que la création d'un système multitâche améliore légèrement la prédiction de TEM tout en générant une prédiction correcte d'informations additionnelles telles que le style de parole, l'accent du locuteur et le type d'émission qui peuvent être des informations complémentaires utiles.

À partir de nos expérimentations, plusieurs perspectives de recherche peuvent être envisagées. Tout d'abord, nous souhaitons améliorer notre approche proposée (CNN) en exploitant des nouveaux types de traits à l'entrée de notre réseau tels que des traits de type : POS, LEX et LM. De plus, nous souhaitons expérimenter des architectures de type réseaux siamois pour apprendre des représentations prenant en compte explicitement des informations sur le style de parole, l'accent du locuteur, et le type d'émission. Ces représentations intermédiaires pourraient être intégrées dans notre système de prédiction pendant la phase d'entraînement afin d'améliorer la qualité du système. Enfin, nous souhaitons aussi étudier l'effet de la quantité des données d'apprentissage Train$_{Pred}$ sur la qualité des systèmes de prédiction, et étudier la robustesse des deux méthodes de prédiction proposées (TR et CNN) lorsqu'elles sont entraînées et/ou évaluées sur des transcriptions automatiques issues d'un autre SRAP.

8. Bibliographie

Asadi A., Schwartz R., Makhoul J., « Automatic detection of new words in a large vocabulary continuous speech recognition system », *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1990.

Belinkov Y., Glass J., « Analyzing Hidden Representations in End-to-End Automatic Speech Recognition Systems », *Advances in Neural Information Processing Systems*, p. 2438-2448, 2017.

Belinkov Y., Màrquez L., Sajjad H., Durrani N., Dalvi F., Glass J., « Evaluating Layers of Representation in Neural Machine Translation on Part-of-Speech and Semantic Tagging Tasks », *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, vol. 1, p. 1-10, 2017.

Chollet F. *et al.*, « Keras », , https ://github.com/fchollet/keras, 2015.

Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., Kuksa P., « Natural language processing (almost) from scratch », *Journal of Machine Learning Research*, vol. 12, nº 8, p. 2493-2537, 2011.

Dai W., Dai C., Qu S., Li J., Das S., « Very deep convolutional neural networks for raw waveforms », *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, p. 421-425, 2017.

De Calmès M., Pérennou G., « BDLEX : a lexicon for spoken and written French », *Proceedings of 1st International Conference on Langage Resources & Evaluation*, p. 1129-1136, 1998.

de Souza J. G. C., Buck C., Turchi M., Negri M., « FBK-UEdin participation to the WMT13 quality estimation shared task », *Proceedings of the Eighth Workshop on Statistical Machine Translation*, p. 352-358, 2013.

de Souza J. G., Zamani H., Negri M., Turchi M., Daniele F., « Multitask learning for adaptive quality estimation of automatically transcribed utterances », *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 714-724, 2015.

Dreschler W. A., Verschuure H., Ludvigsen C., Westermann S., « ICRA noises : artificial noise signals with speech-like spectral and temporal properties for hearing instrument assessment : Ruidos ICRA : Señates de ruido artificial con espectro similar al habla y propiedades temporales para pruebas de instrumentos auditivos », *Audiology*, vol. 40, nº 3, p. 148-157, 2001.

Elloumi Z., Besacier L., Galibert O., Kahn J., Lecouteux B., « ASR PERFORMANCE PREDICTION ON UNSEEN BROADCAST PROGRAMS USING CONVOLUTIONAL NEURAL NETWORKS », *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

Eyben F., Wöllmer M., Schuller B., « Opensmile : The Munich Versatile and Fast Open-source Audio Feature Extractor », *International Conference on Multimedia*, MM '10, ACM, New York, NY, USA, p. 1459-1462, 2010.

Ferreira S., Farinas J., Pinquier J., Rabant S., « Prédiction a priori de la qualité de la transcription automatique de la parole bruitée », *Proc. XXXIIe Journées d'Études sur la Parole*, p. 249-257, 2018.

Galibert O., « Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech. », *in* F. Bimbot, C. Cerisara, C. Fougeron, G. Gravier, L. Lamel, F. Pellegrino, P. Perrier (eds), *Interspeech*, ISCA, p. 1131-1134, 2013.

Galliano S., Geoffrois E., Mostefa D., Choukri K., Bonastre J.-F., Gravier G., « The ESTER phase II evaluation campaign for the rich transcription of French broadcast news. », *Interspeech*, p. 1149-1152, 2005.

Geurts P., Ernst D., Wehenkel L., « Extremely randomized trees », *Machine learning*, vol. 63, n° 1, p. 3-42, 2006.

Gravier G., Adda G., Paulson N., Carré M., Giraudel A., Galibert O., « The ETAPE corpus for the evaluation of speech-based TV content processing in the French language », *LREC-Eighth international conference on Language Resources and Evaluation*, p. na, 2012.

Hermansky H., Variani E., Peddinti V., « Mean temporal distance : Predicting ASR error from temporal properties of speech signal », *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, p. 7423-7426, 2013.

Jalalvand S., Falavigna D., Matassoni M., Svaizer P., Omologo M., « Boosted acoustic model learning and hypotheses rescoring on the CHiME-3 task », *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, IEEE, p. 409-415, 2015a.

Jalalvand S., Negri M., Daniele F., Turchi M., « Driving rover with segment-based asr quality estimation », *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, vol. 1, p. 1095-1105, 2015b.

Jalalvand S., Negri M., Falavigna D., Turchi M., « Driving ROVER with Segment-based ASR Quality Estimation », 01, 2015c.

Jalalvand S., Negri M., Turchi M., de Souza J. G., Falavigna D., Qwaider M. R., « Transcrater : a tool for automatic speech recognition quality estimation », *Proceedings of ACL-2016 System Demonstrations. Berlin, Germany : Association for Computational Linguistics*p. 43-48, 2016.

Jin M., Song Y., Mcloughlin I., Dai L.-R., Ye Z.-F., « LID-senone extraction via deep neural networks for end-to-end language identification », *Proc. of Odyssey*, 2016.

Kahn J., Galibert O., Quintard L., Carré M., Giraudel A., Joly P., « A presentation of the REPERE challenge », *Content-Based Multimedia Indexing (CBMI), 2012 10th International Workshop on*, IEEE, p. 1-6, 2012.

Kim Y., « Convolutional neural networks for sentence classification », *arXiv preprint arXiv :1408.5882*, 2014.

Kingma D. P., Ba J., « Adam : A Method for Stochastic Optimization », *CoRR*, 2014.

Maaten L. v. d., Hinton G., « Visualizing data using t-SNE », *Journal of machine learning research*, vol. 9, n° 11, p. 2579-2605, 2008.

McFee B., Raffel C., Liang D., Ellis D. P., McVicar M., Battenberg E., Nieto O., « librosa : Audio and music signal analysis in python », 2015.

Meinshausen N., Bühlmann P., « Stability selection », *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, vol. 72, n° 4, p. 417-473, 2010.

Meyer B. T., Mallidi S. H., Kayser H., Hermansky H., « Predicting error rates for unknown data in automatic speech recognition », *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, p. 5330-5334, 2017.

Mikolov T., Sutskever I., Chen K., Corrado G., Dean J., « Distributed Representations of Words and Phrases and their Compositionality », *NIPS*, 2013.

Mohamed A.-r., Hinton G., Penn G., « Understanding how deep belief networks perform acoustic modelling », *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, p. 4273-4276, 2012.

Nagamine T., Seltzer M. L., Mesgarani N., « Exploring how deep neural networks form phonemic categories », *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

Negri M., Turchi M., de Souza J. G., Falavigna D., « Quality Estimation for Automatic Speech Recognition. », *COLING*, p. 1813-1823, 2014.

Palaz D., Doss M. M., Collobert R., « Convolutional neural networks-based continuous speech recognition using raw speech signal », *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, p. 4295-4299, 2015.

Pellegrini T., Mouysset S., « Inferring phonemic classes from CNN activation maps using clustering techniques », *Interspeech*, p. pp-1290, 2016.

Piczak K. J., « Environmental sound classification with convolutional neural networks », *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, IEEE, p. 1-6, 2015.

Povey D., Ghoshal A., Boulianne G., Burget L., Glembek O., Goel N., Hannemann M., Motlicek P., Qian Y., Schwarz P. *et al.*, « The Kaldi speech recognition toolkit », *IEEE 2011 workshop on automatic speech recognition and understanding*, IEEE Signal Processing Society, 2011.

Sainath T. N., Weiss R. J., Senior A., Wilson K. W., Vinyals O., « Learning the speech front-end with raw waveform CLDNNs », *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

Schmid H., « Treetagger| a language independent part-of-speech tagger », *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, vol. 43, p. 28, 1995.

Shi X., Padhi I., Knight K., « Does string-based neural MT learn source syntax ? », *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, p. 1526-1534, 2016.

Stolcke A. *et al.*, « SRILM-an extensible language modeling toolkit. », *Interspeech*, vol. 2002, p. 2002, 2002.

Wang S., Qian Y., Yu K., « What Does the Speaker Embedding Encode ? », *Interspeech*, vol. 2017, p. 1497-1501, 2017.

Wu Z., King S., « Investigating gated recurrent neural networks for speech synthesis », *CoRR*, 2016.

Young S. R., « Recognition Confidence Measures : Detection of Misrecognitions and Out-Of-Vocabulary Words », *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, p. 21-24, 1994.

Zeiler M. D., « ADADELTA : An Adaptive Learning Rate Method », *CoRR*, 2012.

Adversarial Networks for Machine Reading

Quentin Grail — Julien Perez — Tomi Silander

NAVER LABS Europe,
6-8, chemin de Maupertuis, 38240 Meylan, France

ABSTRACT. *Deep machine reading models have recently progressed remarkably with the help of differentiable reasoning models. In this context, deep end-to-end trainable networks enhanced with memory and attention have demonstrated promising performance on simple natural language based reasoning tasks. However, the training of machine comprehension models commonly requires a large annotated question-answer dataset for learning. In this paper, we explore the paradigm of adversarial learning and self-play for machine reading comprehension. Inspired by the success in the domain of game learning, we propose a novel approach to train machine comprehension models based on a coupled attention-based model. In this approach, a reader network is in charge of finding answers to the questions regarding a passage of text, while an obfuscation network tries to obfuscate spans of text in order to minimize the probability of success of the reader. The model is evaluated on several question-answering corpora. The proposed learning paradigm and associated models show promising results.*

RÉSUMÉ. *Les modèles d'apprentissage profond utilisés sur des tâches de lecture automatique ont remarquablement progressé ces dernières années. Parmi ces architectures, les modèles d'attention et à mémoire ont démontré des performances encourageantes sur différentes tâches de raisonnement. Cependant, le protocole d'apprentissage de ces modèles suppose qu'une grande quantité d'exemples soient disponibles. Dans cet article, nous proposons d'exploiter l'apprentissage adversarial et le self-play durant la phase d'entraînement des modèles. Nous proposons un nouveau protocole d'apprentissage sous la forme d'un jeu entre deux modèles adverses. Ces modèles sont mis en compétition sur une tâche de question-réponse. D'un côté un modèle, appelé « lecteur », est chargé de répondre à une question portant sur un document, et de l'autre un modèle, appelé « réseau d'obfuscation », est chargé d'obfusquer un passage du document de manière à maximiser la probabilité de tromper le lecteur sur ce document corrompu. Nous avons testé ce protocole sur plusieurs datasets de question-réponse, et ce nouveau protocole d'apprentissage adversarial permet d'obtenir des résultats encourageants.*

KEYWORDS: *Machine reading, adversarial learning, deep learning.*

MOTS-CLÉS : *Modèles de lecture, apprentissage adversarial, apprentissage profond.*

1. Introduction

Automatic comprehension of text is one of the main goals of natural language processing. While the ability of a machine to understand text can be assessed in many different ways, several benchmark datasets have recently been created to focus on answering questions as a way to evaluate machine comprehension (Richardson *et al.*, 2013; Hermann *et al.*, 2015; Hill *et al.*, 2015; Weston *et al.*, 2015; Rajpurkar *et al.*, 2016; Nguyen *et al.*, 2016). In this setup, a piece of text such as a news article or a story is presented to the machine. The machine is then expected to answer one or multiple questions related to the text. Figure1 presents question-answering example from the Cambridge dataset. Solving this task provides tools that help users to efficiently access large amounts of information. Furthermore, it also acts as an important proxy task to assess models of natural language understanding and reasoning. Recently, publication of many large datasets (Hermann *et al.*, 2015; Rajpurkar *et al.*, 2016; Trischler *et al.*, 2017; Nguyen *et al.*, 2016) have contributed to significant advancement in machine comprehension and question-answering. Recent neural models for machine comprehension are now approaching human comprehension on some of these benchmarks, and there is currently a lot of novel and promising research on parametric models that feature reasoning capabilities using techniques such as attention and memory. The work in this field is currently following the paradigm of supervised learning which makes it strictly dependent on the availability of annotated datasets; the production of which is costly. Since the 1990s an increasingly common research activity has been dedicated to self-play and adversariality to overcome this dependency and allow a model to exploit its own decisions to improve itself. Some famous examples are related to policy learning in games. TD-Gammon (Tesauro, 1995) was a neural network controller for backgammon which achieved near top player performance using self-play as learning paradigm. More recently, DeepMind's AlphaGo (Silver *et al.*, 2016) used the same paradigm to win against the currently best human Go player. The major advantage of such a setting is to alleviate the learning procedure's dependency on an available annotated dataset. Two models can be set up to learn and improve their performance by acting one against the other in so-called sparring patterns.

In this paper, we adapt this paradigm to the domain of machine reading. On the first hand, a **reader network** is trained to learn to answer questions regarding a passage of text. On the other hand, an **obfuscation network** learns to obfuscate words of a given passage in order to minimize the probability of the reading model to successfully answer the question. We developed a sequential learning protocol in order to gradually improve the quality of the models. This paradigm separates itself from the current approach of joint question and answer learning from text as proposed by Wang *et al.* (2017a). Indeed, rather than using question generation as regularizer of a reader model, we suggest using adversarial training to free us from the constraint of strict and bounded supervision and to enhance the robustness of the answering model.

Our contributions can be summarized as follows: (1) We propose a new learning paradigm for machine comprehension based on adversarial training. (2) With exper-

Document: *This is a very reasonably priced hotel of a good standard for a short visit. I had a single room (407) at the front of the hotel. On the negative side, the room was very small, there is street and aircraft noise and it was too warm. However, they have made excellent use of the space available and the decor was good. The bathroom was newly fitted out and the shower was excellent. I found the staff efficient and friendly. Much better than the last place I stayed in London and cheaper. I didn't have breakfast but it was reasonably priced.*
Question: *How is the service?*
Answer: *3/5.*

Figure 1. *An example from the TripAdvisor dataset.*

iments in several machine reading corpora and with several neural architectures, we show that this methodology allows us to overcome the requirement of strict supervision and provides robustness to noise in question answering. (3) The attention mechanism allows the visualization of passages considered as meaningful by the obfuscation network. We present the results of this attention mechanism on multiple examples.

Roadmap:
In Section 2 we review the state-of-the-art of machine reading comprehension, the paradigm of adversarial learning and its relation to the adversarial learning protocol proposed in this article. In Section 3, we formalize our adversarial learning protocol and introduce the two types of architectures used in this work. In Section 4 we introduce the corpora used for evaluation. In Section 5 we present our experimental results, and finally in Section 6, we demonstrate several visualizations of the decisions and attention values produced by the coupled models.

2. Related work

2.1. *End-to-end machine reading*

The task of end-to-end machine reading consists of learning, in a supervised manner, to answer a question given a passage of text. One of the popular formal settings of the problem is the cloze-style question-answering task. This task involves tuples of the form (d, q, a, C), where d is a document (context) and q is a query over the contents of d, in which a word has been replaced with a placeholder. The objective is to fill the placeholder with a word chosen among the set of candidates C. The correct answer is a. In this work, we consider datasets where each candidate $c \in C$ has at least one token which also appears in the document. The task can then be described as: given a document-query pair (d, q), find $a \in C$ which answers q. Below we provide an overview of representative neural network architectures which have been applied to this problem.

LSTMs with Attention: Several architectures introduced in Hermann *et al.* (2015) employ LSTM units to compute a combined document-query representation $g(d, q)$, which is used to rank the candidate answers. These include the DeepLSTM Reader which performs a single forward pass through the concatenated (document, query) pair to obtain $g(d, q)$; the Attentive Reader which first computes a document vector $d(q)$ by a weighted aggregation of words according to attentions based on q, and then combines $d(q)$ and q to obtain their joint representation $g(d(q), q)$; and the Impatient Reader where the document representation is built incrementally. The architecture of the Attentive Reader has been simplified recently in Stanford Attentive Reader, where shallower recurrent units were used with a bilinear form for the query-document attention (Chen *et al.*, 2016).

Attention-Sum Reader: The Attention-Sum (AS) Reader (Kadlec *et al.*, 2016) uses two bidirectional GRU networks to encode both d and q into vectors. A probability distribution over the entities in d is obtained by computing dot products between q and the entity embeddings and taking the softmax. Then, an aggregation scheme called "pointer-sum attention" is further applied to sum the probabilities of the same entity, so that frequent entities in the document will be favored compared to rare ones. Building on the AS Reader, the Attention-over-Attention (AoA) Reader (Cui *et al.*, 2017) introduces a two-way attention mechanism where the query and the document are mutually attentive to each other.

Multi-hop Architectures: Memory Networks (MemNets) were proposed in Weston *et al.* (2014), where each sentence in the document is encoded to a memory cell by aggregating nearby words. Attention over the memory slots given the query is used to compute an overall attention and to renew the query representation over multiple iterations, allowing certain types of reasoning over the salient facts in the memory and the query. Neural Semantic Encoders (NSE) (Yu and Munkhdalai, 2017) extended MemNets by introducing a write operation which can evolve the memory over time during the course of reading. Iterative reasoning has been found effective in several more recent models, including the Iterative Attentive Reader (Sordoni *et al.*, 2016) and ReasoNet (Shen *et al.*, 2016). The latter allows dynamic reasoning steps and is trained with reinforcement learning.

In other related work, EpiReader (Trischler *et al.*, 2016) consists of two networks, where one proposes a small set of candidate answers, and the other reranks the proposed candidates conditioned on the query and the context. Bi-Directional Attention Flow network (BiDAF) (Seo *et al.*, 2016) adopts a multi-stage hierarchical architecture along with a flow-based attention mechanism.

2.2. *Adversarial learning*

The idea of using an adversarial learning protocol has been very popular during the last couple of years, particularly in the field of generative models. Indeed Generative Adversarial Networks (GANs), introduced in (Goodfellow *et al.*, 2014a), have now

lots of applications and allowed the training protocol to go beyond the strict supervision of the answer. The main principle of Generative Adversarial Networks (GANs) is to train jointly two adversarial models. These two models are challenging each other with opposing objectives and jointly progressing in the task they are designed for. In machine reading, it has been recently observed that answering a question regarding a text passage and predicting the question regarding a text passage are interesting tasks to model jointly. Consequently, several papers have proposed using the question generation as a regularization task to improve the passage encoding model of a neural reader (Yuan *et al.*, 2017; Wang *et al.*, 2017a). In this paper, we acknowledge that these two tasks may indeed be complementary but we believe adversarial training in two player games will lead to similar advantages than those observed previously. As generating a question for a passage is hard, we adapt recent work by Guo *et al.* (2017) and define the learning of an obfuscation network as a complementary task to the task of learning a reader. Such an obfuscation network tries to find the most meaningful spans of text to obfuscate in a given passage for a given question in order to minimize the probability of the reader successfully answering the question.

2.3. *Adaptive dropout*

Several studies have recently featured the idea of challenging deep machine reading models with adversarial examples (Miyato *et al.*, 2016; Jia and Liang, 2017). While this kind of approach is well known in computer vision (Goodfellow *et al.*, 2014b), it seems to be relevant also for natural language processing. More precisely, Jia and Liang (2017) demonstrated that a large majority of the recent state-of-the-art deep machine reading models suffer from a lack of robustness regarding adversarial examples because of their oversensitivity. It means that small perturbation in the input can completely disturb the model. In these studies, models suffer from the so-called *catastrophic forgetting*; their average accuracies were decreased by half when tested on corrupted data, i.e., on documents with an additional sentence at the end, which normally should not affect the answer.

One of the attempts to prevent overfitting is to randomly drop network units while training (Srivastava *et al.*, 2014). Such an approach effectively results in combining many different neural networks to make a prediction. In the same spirit, training a model on a dataset with corrupted data is shown to decrease overfitting. Maaten *et al.* (2013) suggest different ways to corrupt a document, for example by adding noise into the input features; our work refers to what they call the *blankout corruption*, which consist of randomly deleting features in the input documents (texts or images in this case) with probability q. However, learning only from predefined adversarial examples appears sub-optimal since it is not dynamically adapted to the performance of the reader.

We think random corruption is not the most efficient way to corrupt the data, but that the corruption should be dynamically adapted to the performance of the reader. While obfuscation of one of the keywords can be too hard for the reader at the begin-

ning of the training, obfuscation of a meaningless word is unlikely to have any effect on the reader that is good enough. The learning protocol we propose aims to handle this by training jointly the obfuscation network and the reader in order to adapt the corruption difficulty to the reader's performance.

3. Adversarial reading networks

The model we propose is built to use this kind of adversariality as an adaptive dropout by challenging the reader with more and more difficult tasks during the learning. Indeed, we utilize *asymmetric self-play* to train a model called an *obfuscation network* that plays an adversarial game against a *reader*. The obfuscation network is acquiring knowledge about the reader's behaviour during the training, and it generates increasingly hard adversarial examples. Beyond increasing artificially the size of the available dataset, this adaptive behaviour of the obfuscation network prevents catastrophic forgetting phenomena of the reader. In this section, we first explain our protocol of adversarial training for robust machine comprehension and then describe the reader and obfuscation network models.

3.1. *Adversarial learning protocol*

The overall framework is a turn-based question-answering game described in Figure 2 and algorithm 1. At the beginning of each round t, the obfuscation network obfuscates one word for each document sampled from the training corpus. We fix the ratio of corrupted data / clear data to a ratio $\lambda \in [0, 1]$ of the dataset. Indeed, too low a percentage of corrupted data might not have any effect on the training and a too high one will prevent the reader of learning well. The reader is then trained on a subset of this obfuscated corpus and tested on the remaining subset. Note that both train and test sets contain corrupted data. Finally, the obfuscation network gets back a set of rewards regarding the reader performance on the obfuscated stories. Given a tuple $(d, d^\dagger, q)$ where d is the original document, $d^\dagger$ the document with an obfuscated word proposed by the obfuscation network and q the associated question, the reward r given to the obfuscation network is defined as follows:

$$r = \begin{cases} 1 & \text{if the reader answers well on } d \text{ and fail on } d^\dagger \\ 0 & \text{otherwise.} \end{cases}$$

The reward given to the obfuscation network is a direct measurement of the impact of the obfuscation on the reader performance. All the previously collected rewards are stored and used for experience replay throughout the turns. After each learning turn, all the parameters of the obfuscation network are reinitialized and retrained on all the recorded rewards. Throughout the turns, the obfuscation network accumulates information about the reader behaviour and proposes more challenging tasks as the game continues. Among the corrupted documents that the obfuscation network proposes

to the reader, 80% of the documents maximize the probability of fooling the reader from the obfuscation network point of view and 20% are randomly corrupted in order to ensure exploration. Finally, the reader keeps improving through time and any catastrophic forgetting is compensated at the next turn of the obfuscation network by focusing on these errors.

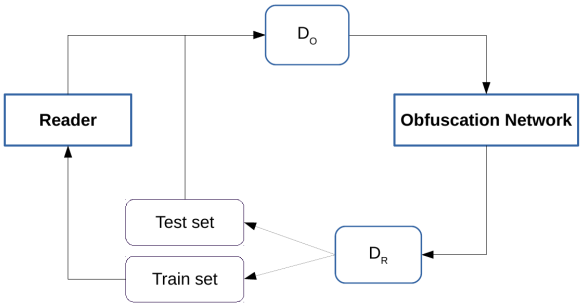


Figure 2. *Adversarial learning protocol with $D_R = \{d_i, q_i, a_i\}_i$ the reader dataset composed by tuples (document, question, answer) and $D_O = \{d_i, q_i, a_i, r_i\}_i$ the obfuscation network dataset composed by tuples (document, question, answer, reward from the reader).*

To more formally specify loss functions for the reader and the obfuscation network, let $\widehat{a}_{ij} \triangleq P(ans_{ij}|q_i, d_i^\dagger)$ denote the reader's predictive probability for ans_{ij} being the correct answer to the question q_i for $j \in [0, n]$ where n is the number of possible answers. Let us denote the index of the actual correct answer by ij^*. The reader is trained to minimize the cumulative log-loss (cross-entropy) for N questions

$$\mathscr{L}_{\text{Reader}} = -\sum_{i=1}^{N} \log \widehat{a}_{ij^*}. \qquad [1]$$

The obfuscation network is trained to fool the reader, so it suffers a loss when it fails to predict whether the reader gives a correct answer ans_{ij^*}. By denoting the indicator of the reader answering the question q_i wrong by $fail_i \in \{0, 1\}$ and obfuscation network's estimate of the probability of this failure by $\widehat{a}_i \triangleq P(fail_i = 1|q_i, d_i^\dagger)$, the obfuscation network's loss is defined as

$$\mathscr{L}_{\text{ObfNet}} = -\sum_{i=1}^{N} fail_i \log \widehat{a}_i + (1 - fail_i) \log(1 - \widehat{a}_i). \qquad [2]$$

Algorithm 1 Adversarial training

input: Let I be the initial set of data $\{(d, q, a)\}_i$ where d, q, a are sequences of index representing a document, a question and an answer.
Let A be the training set (80% of I)
Let B be the validation set (10% of I)
Let C be the testing set (10% of I)
Let D be an empty dataset
$t = 0$
while $t <$ NB_MAX_EPOCHS **do**
  Split A into A_1 (80%) and A_2 (20%)
  if t = 0 **then**
    Let $A_1^\dagger$ be A_1 with 20% of random corruption
    Let $A_2^\dagger$ be A_2 with 100% of random corruption
  else
    Reinitialize all the parameters of the obfuscation network
    Train the obfuscation network on D
    Let $A_1^\dagger$ be A_1 with 20% of data corrupted by the obfuscation network
    Let $A_2^\dagger$ be A_2 with 100% of data corrupted by the obfuscation network
  end if
  Train one epoch of the reader on $A_1^\dagger$
  for all $((d, q, a) \in A_2, (d^\dagger, q, a) \in A_2^\dagger)$ **do**
    Let r be the reward given to the obfuscation network
    if the reader succeed on d and fails on $d^\dagger$ **then**
      $D \leftarrow \{D \cup (d^\dagger, q, a, r = 1)\}$
    else if the reader succeed on d and succeed on $d^\dagger$ **then**
      $D \leftarrow \{D \cup (d^\dagger, q, a, r = 0)\}$
    end if
  end for
  Let ε_t be the empirical error of the reader on B
  if $\varepsilon_t > \varepsilon_{t-1}$ **then**
    Stop the learning
  end if
  t $\leftarrow$ t + 1
end while
Report the empirical error of the reader on C

3.2. *Baseline learning protocol*

In our reference protocol, the corruption is made by randomly obfuscating a word in several documents. This is a naive variation of the first protocol where the obfuscation network does not learn from the reader feedback at all. In fact, this protocol is similar to a dropout regularization on the embeddings layer that allows avoiding overfitting the training set. However, the obfuscation is independent of the reader per-

formance; especially, it does not take into account the difficulty of the questions. In practice, this simple adversarial protocol still improves the robustness of the results compared to a standard learning protocol. This learning protocol has strong similarities with the one proposed by Maaten *et al.* (2013).

3.3. *Reader network*

To illustrate this work, we investigate two types of neural architectures: a memory based architecture with a Gated End-to-End Memory Network (Liu and Perez, 2017b) (`GMemN2N`) and a multi-layer attention based architecture largely inspired by the recent `R-Net`(Wang *et al.*, 2017b) excepted for its output layer, adapted to the format of the datasets used in this work. These two architectures are state-of-the-art models for machine reading and most of the recent models are a combination of layers included in these two architectures. Paragraphs below describe these two architectures and how we have integrated them in the adversarial learning protocol.

3.3.1. *Gated End-to-End Memory Network reader*

The first model used as a reader is a Gated End-to-End Memory Network (Liu and Perez, 2017b), `GMemN2N` (Figure 3). This architecture is based on two different memory cells and an output prediction. An input memory representation $\{m_i\}$ and an output representation $\{c_i\}$ are used to store embedding representations of inputs. Suppose that an input of the model is a tuple (d, q) where d is a document, i.e., a set of sentences $\{s_i\}$, and q is a query about d. The entire set of sentences is converted into input memory vectors $m_i = A\Phi(s_i)$ and output memory vectors $c_i = C\Phi(s_i)$ by using two embedding matrices A and C. The question q is also embedded using a third matrix B, $u = B\Psi(q)$ of the same dimension as A and C, where Φ and Ψ are respectively the document embedding function and the question embedding function described in the next paragraph. The input memory is used to compute the relevance of each sentence in its context regarding the question, by computing the inner product of the input memory sentence representation with the query. A softmax is then used to map the inner product to a probability. The response $o = \sum_i p_i c_i$ from the output memory is the sum of the output memory vectors $\{c_i\}$ weighted with the sentence relevancies calculated before $p_i = \text{softmax}(\boldsymbol{u}^T m_i)$. A gated mechanism is used when we update the value of the controller u:

$$T^k(u^k) = \sigma(W_T^k u^k + b_T^k), \quad [3]$$

$$\mathbf{u^{k+1}} = o^k \odot T^k(u^k) + u^k \odot (1 - T^k(u^k)), \quad [4]$$

where W_T^k are matrices of size $d \times d$ and b_T^k a vector of size d with d the size of the memory cells.

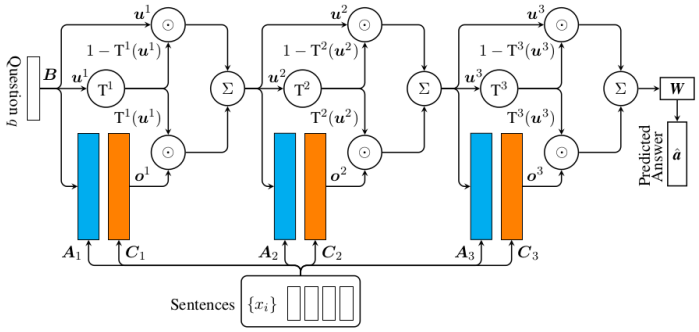


Figure 3. *Gated End-to-End Memory Network (Liu and Perez, 2017b).*

Assuming we use a model with K hops of memory, the final prediction is:

$$\widehat{a} = \text{softmax}(W(o^K + u^K) + b), \quad [5]$$

where W is a matrix of size $d \times v$ and b a vector of size d with v the number of candidate answers. In this model, we do not use the adjacent or layer-wise weight tying scheme and all the matrix A^k and B^k of the multiple hops are different.

Text and question representations (Figure 4): To build the sentence representations, we use a 1-dimensional Convolutional Neural Network (CNN) with a list of filter sizes over all the sentences as proposed in Kim (2014). Let $[s_1, \ldots, s_N]$ be the vectorial representation of a document with N sentences where $s_i = [w_{i,1}, w_{i,2}, \ldots, w_{i,n}]$ is the $i-th$ sentence which contains n words. Given a convolutional filter $F \in \mathbb{R}^{h \times d}$ where h is the width of the convolutional window, i.e, the number words it overlaps, the convolutional layer produces:

$$c_{i,j} = f(F \odot [Ew_{i,j}, \ldots, Ew_{i,j+h}]), \forall j \in [1, n-j], \quad [6]$$

where $\odot$ is the element-wise multiplication, f a rectified linear unit (ReLU) and E is the embedding matrix of size $d \times V$ where V is the vocabulary size and d the word embedding size. Then, a max pooling operator is applied to this vector to extract features. Given a filter F, after a convolutional operation and a max pooling operation, we obtain a feature $\hat{c}_i = \max_j(c_{i,j})$ from the i^{th} sentence of the text. Multiple filters with varying sizes are used. Assume that our model uses N_s different filter sizes and N_f for each size, we are able to extract $N_s \times N_f$ features for one sentence. The final representation of the sentence is the concatenation of the extracted features from all the filters:

$$\Phi(s_i) = [\hat{c}_{iF_1}, \hat{c}_{iF_2}, \ldots, \hat{c}_{iF_{N_s * N_f}}]. \quad [7]$$

Compared to an LSTM encoding the CNN layer is faster and gives better results on the different tasks we evaluated our model. This result seems coherent with recent

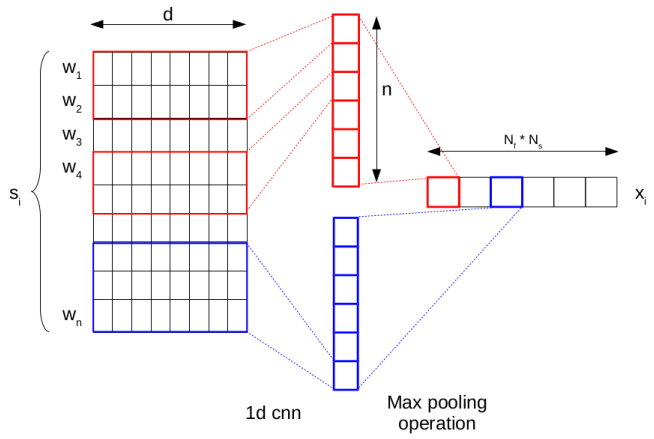


Figure 4. *An encoded sentence where d is the word embedding size, N_f the number of filters of each size and N_s the number of different filter sizes used.*

results of Dauphin *et al.* (2017). We use a bidirectional GRU to encode the question. The question representation $\Psi(q)$ is the concatenation of the final states of the forward and backward GRU on this question.

3.3.2. *R-Net based network*

The second architecture investigated in this article is based on the state-of-the-art `R-Net` model (Wang *et al.*, 2017b). The main part of the architecture remains the same as the original model, except for the last layer. We replaced the pointer network, originally used to select in the document the span of text that corresponds to the answer, by a fully connected layer followed by a softmax to output the probability of each candidate word to be the answer. The following lines describe the structure of this architecture, composed of multiple stacked layers.

Encoding layer: Each sentence is tokenized by word and each token is represented by the concatenation of the word level, and character level embeddings. The word level embedding is computed via a lookup table initialized with `GloVe` pre-trained embeddings and the character embedding of a token is the final state of a GRU network over the sequence of its characters. Finally, these tokens are fed to a Recurrent Neural Network (RNN) and the document and question are represented by the intermediate states of this RNN.

Gated question/document attention: Assuming that $d = \{u_i^d\}_{i=0}^N$ and $q = \{u_i^q\}_{i=0}^N$ are the sequences of embedding tokens of the document and the question after the encoding layer with N the length of the document and n the length of the

question. Then we compute an attention between the representation of the question and each token of the document. The document is transformed to $d = \{v_i^d\}_{i=0}^N$ with:

$$v_i^d = RNN(v_{i-1}^d, [u_i^d, c_i]),$$

where c_i is an attention vector of the question over the token i of the document. This layer produces a question-aware representation of the document.

Self-attention: So far each token contains information from the question due to the question/document attention layer and from its surrounding context due to the RNN at the end of the encoding part but does not handle long-term dependencies inside the document. The self-attention layer produces an attention between the whole document and each individual token of it. $d = \{h_i^d\}_{i=0}^N$ with:

$$h_i^d = BiRNN(h_{i-1}^d, [v_i^d, c_i]),$$

where c_i is an attention vector of the whole document over the token i.

Output layer: The decision support is the concatenation of the h_i, for $i \in [0, N]$. $o^d = \text{concat}(\{h_i\}_{i=0}^N)$ and finally:

$$\widehat{a} = \text{softmax}(Wo^d + b),$$

where W is a matrix of size $N * d \times v$ and b a vector of size d with v the number of candidate answers.

3.4. *Obfuscation network*

The objective of this model is to predict the probability of the reader to successfully respond to a question about a document with an obfuscated word. This estimate will be used by the obfuscation network to determine the position of the obfuscated word in the document which maximizes the probability of the reader to fail its task. We use a similar architecture as the reader, i.e a `GMemN2N` when the reader is a `GMemN2N` and a `R-Net` when the reader is a `R-Net`. However, on the last layer, a sigmoid function is used to predict the probability of the reader to fail on this input: Assuming that o is the decision support of the obfuscation network, then:

$$\widehat{a} = \sigma(Wo + b), \qquad [8]$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ and $\widehat{a} \in [0, 1]$ is the predicted probability of failure of the reader and W a matrix of size $d \times 1$. We impose this symmetry between the architecture of the reader and of the obfuscation network in order to keep a fair challenge between the two adversary networks.

An input to the reader is a tuple $(d^\dagger, q)$ where $d^\dagger$ is a document with an obfuscated word. To obfuscate a word, we replace it by the word *unk* for *unknown*. The output of the obfuscation network is a real number $r \in [0, 1]$ which is the expected probability of the reader to fail on the question. The objective of the obfuscation network is

to select the corrupted document which maximizes this reward. We use the same text passage and query representation as for the reader, based on a CNN with different filter sizes for the document and the two last hidden states of a bidirectional Gated Rectified Unit (GRU) recurrent network for the question encoding for the `GMemN2N` and based on character and word level embeddings for the `R-Net`. Both models are fully differentiable.

4. Datasets and data preprocessing

Cambridge Dialogs: The transactional dialog corpus proposed by Rojas-Barahona *et al.* (2017) has been produced by a crowdsourced version of the Wizard-of-Oz paradigm. It was originally designed for dialog state tracking, but Liu and Perez (2017a) have shown that this task could also be considered as a reading task. In such setting, the informable slots provided as metadata to each dialog were used to produce questions for a dialog comprehension task. The dataset deals with an agent assisting a user to find a restaurant in Cambridge, UK. To propose the best matching restaurant, the system needs to extract 3 constraints which correspond to the informable slots in the dialog state tracking task: *Food, Price range, Area*. Given a dialog between an agent and a user, these informable slots become questions for the model we propose. The dataset contains 680 different dialogs about 99 different restaurants. We preprocess the dataset to transform it into a question-answering dataset by using the three informable slot types as questions about a given dialog. After this preprocessing operation, we end up with our question-answering formatted dataset which contains 1,352 possible answers.

Document: *I want the phone number of a moderately priced restaurant with Spanish food.* *La Tasca would fit the bill. Its phone number is 01223 464630.* *Can you tell me what area of town it is located?* *La Tasca is located in the center part of town.* *Thank you, goodbye.* *You're welcome.* **Question:** *What is the area?* **Answer:** *Center.*

Table 1. *An example from the Cambridge dataset formatted for question-answering task.*

TripAdvisor aspect-based sentiment analysis: This dataset contains a total of 235K detailed reviews extracted from the TripAdvisor website and originally released by Wang *et al.* (2010). These reviews represent around 1,850 hotels. Each review is associated to an overall rating, between 0 and 5 stars. Furthermore, 7 aspects: *value, room, location, cleanliness, checkin/front desk, service,* and *business service* are available. We transform the dataset into a question-answering task over a given review. Concretely, for each review, a question is an aspect and we use the number of

stars as the answer. This kind of machine-reading approach to sentiment analysis was previously proposed in Tang *et al.* (2016).

Document: *Service was ok, staff helpful, room was basic, marks on bedding top cover looked like blood, sheets clean, bathroom not so nice, broken tiles on floor, shower head was disgusting and needed to be replaced, location was good, close to the metro and the Colosseum, both only a 10 min walk, liked that the hotel was close to many cafe's restaurant's, disliked the shower in room.*

Question: *How is the cleanliness?*
Answer: 2/5.

Question: *How is the service?*
Answer: 3/5.

Table 2. *An example from the TripAdvisor dataset.*

Children's Book Test (CBT): The dataset is built from freely available books (Hill *et al.*, 2015) produced by Project Gutenberg [1]. The training data consists of tuples (S, q, C, a) where S is the *context* composed of 20 consecutive sentences from the book, q is the *query*, C a set of 10 *candidate answers* and a the *answer*. The query q is the 21^{st} sentence, i.e., the sentence that directly follows the 20 sentences of the *context* and where one word is removed and replaced by a missing word symbol. Questions are grouped into 4 distinct categories depending of the type of the removed word: Named Entities (NE), (Common) Nouns (CN), Verbs (V) and Prepositions (P). This division of answers according to the type of the word that has been removed give a way to evaluate the performance of a model in different situations. It provides relevant information on the strengths and weaknesses of a given architecture. The training contains 669,343 inputs (context+query) and we evaluated our models on the provided test set which contains 10,000 inputs, 2,500 per category. This dataset evaluates the capability that a model has to predict a word based on its context.

1. `https://www.gutenberg.org`.

Document:
1 When she got home she shut herself up in her room and cried.
2 There was nothing for her to do but resign, she thought dismally.
3 On the following Saturday Esther went for an afternoon walk, carrying her Kodak with her.
4 It was a brilliantly fine autumn day, and woods and fields were basking in a mellow haze.
19 Bob and Alf Cropper were up among the boughs picking the plums.
20 On the ground beneath them stood their father with a basket of fruit in his hand.

Question: *21 Mr. Cropper looked at the **XXXXX** and from it to Esther.*
Answer: *proof*
Candidates: *Saturday* | *boughs* | *face* | *father* | *home* | *nothing* | *proof* | *remarks* | *smile* | *woods*

Table 3. *An example from the CBT dataset.*

5. Experiments

In this section, we present our experimental settings and the results of this adversarial training protocol on the three datasets presented in Section 4.

5.1. *Training details*

10% of the dataset was randomly held-out to create a test set. We split the dataset before all the training operations and each protocol was tested on the same test dataset. For the training phase, we split the training dataset to extract a validation set to perform early stopping. We used Adam optimizer (Kingma and Ba, 2014) with a starting learning rate of 0.0005. We set the dropout to 0.9 which means that during training, randomly selected 10% of the parameters are not used during the forward pass and not updated during the backward propagation of error. We also added the gated memory mechanism (Liu and Perez, 2017b) that dynamically regulates the access to the memory blocks. This mechanism had a very positive effect on the overall performance of our models. All weights were initialized randomly from a Gaussian distribution with zero mean and a standard deviation of 0.1. We augmented the loss with the sum of squares of the model parameters.

The hyperparameters have been chosen via cross-validation on the validation set of the different datasets. We set the batch size to 16 inputs and we used word embeddings of size 300. We initialized all the embedding matrices with pre-trained `GloVe` word vectors (Pennington *et al.*, 2014) and used random vectors for the words not present in

the GloVe. It seems that for our experiments CNN encoding does not improve only the overall accuracy of the model compared to LSTM but also the stability by decreasing the variance of the results. So, in practice, we used 128 filters of size 2, 3, 5 and 8 resulting in a total of 512 filters for the one-dimensional convolutional layer.

We repeated each training 10 times for the first two datasets and report maximum and average accuracy. The average value corresponds to the average score over the 10 runs on the test set. Maximum value corresponds to the score on the test set achieved by the model that performed best on the validation set. During the adversarial learning, the dataset contained 70% of clear dialogs and 30% of corrupted dialogs, $\lambda = 0.3$. Inside these corrupted data, 20% were randomly obfuscated by the obfuscation network in order to make it learn from exploration and the obfuscation network maximized its reward for the remaining 80%. Due to the format of the dataset, we slightly modified the output layer of our reader for the CBT task. Instead of projecting on a set of candidate answers, the last layer of the reader made a projection on the entire vocabulary $\widehat{\boldsymbol{a}} = \sigma(M \odot W(o^K + u^K))$ where W is a matrix of size $V \times d$ with V the vocabulary size, $\odot$ the elementwise product and M the mask vector of size V containing 1 if the corresponding word is proposed in the candidate answers, and 0 otherwise.

5.2. *Results*

In this section, we report the results of our implementation of two baselines: a simple logistic regression and an Attention-Sum Reader (Kadlec *et al.*, 2016). Then we present the results of our implementation of the two neural architectures presented in Section 3.3, trained with the standard training, the *uniform* training, which is the reader trained with the baseline protocol 3.2 and with our adversarial learning protocol 3.1.

	Log Reg	ASR	GMemN2N			uniform GMemN2N			adversarial GMemN2N		
hops			4	5	6	4	5	6	4	5	6
Max	58.4	40.8	82.1	85.8	80.6	85.1	85.8	82.8	82.8	79.8	**88.1**
Mean	58.2	39.5	76.9	74.8	74.2	77.4	77.7	74.9	**79.8**	77.8	79.6

	R-Net	uniform R-Net	adversarial R-Net
Max	88.1	89.5	**90.8**
Mean	87.5	89.2	**90.0**

Table 4. *Average and maximum accuracy (%) on the Cambridge dataset on 10 replications. In bold, the best result per architecture.*

	Log Reg	ASR	GMemN2N			uniform GMemN2N			adversarial GMemN2N		
hops			4	5	6	4	5	6	4	5	6
Max	59.4	45.2	62.3	62.4	60.5	63.1	61.4	63.1	**64.6**	63.5	62.3
Mean	59.0	42.3	60.8	60.6	58.5	62.3	60.3	59.6	**62.8**	61.2	60.8

	R-Net	uniform R-Net	adversarial R-Net
Max	62.3	63.8	**64.5**
Mean	61.9	62.2	**63.0**

Table 5. *Average and maximum accuracy (%) on the TripAdvisor dataset on 10 replications. In bold, the best result per architecture.*

Tables 4 and 5 display the scores obtained by these models on the Cambrige and TripAdvisor datasets. Each experiment was run 10 times and we report in this table the maximum score on the test set (based on the validation set) and the average score. The precise number of hops needed to achieve the best performance with the GMemN2N is not obvious, so we present all the results for readers and obfuscation networks between 4 and 6 hops.

We observe that the **adversarial learning protocol improves the accuracy** of the GMemN2N and R-Net compared to the standard and uniform training protocol for all the experiments.

We improve the score of the reader by 2.3 points on the Cambridge task for a GMemN2N with 6 hops compared to the standard training. This adversarial protocol, applied to the R-Net architecture, improves the average score by 2.5 points on this dataset.

The best performance on the TripAdvisor dataset was achieved by the adversarial R-Net. On 10 replications of the experiment, the average accuracy of this model was improved by 1.1 points compared to the standard approach.

The GMemN2N with 4 hops achieved the best performance of this architecture. The accuracy was improved by 1.5 points when the model was trained with our adversarial protocol.

The uniform protocol improves the stability of the performance compared to a standard reader but further improvements were obtained with the adversarial protocol which improved both the overall accuracy and the stability of the performance. Indeed the variance of the results decreased when the training was done with the adversarial protocol, especially for the GMemN2N. Such architecture does not always converge to the optimal minima and the adversarial learning, acting as an adaptive dropout, seems to help the model to generalized better. It is not clear, for this task, whether the number of hops, between 4 and 6, affects the general behaviour, but we achieved the best performance with our adversarial protocol and a reader with 6 hops.

	Log Reg				ASR			
Task	P	V	NE	CN	P	V	NE	CN
Max	56.3	37.1	26.5	25.6	24.7	32.7	22.1	18.3

	GMemN2N				uniform GMemN2N				adversarial GMemN2N			
Task	P	V	NE	CN	P	V	NE	CN	P	V	NE	CN
Max	56.0	58.5	31.9	39.0	58.1	53.6	31.6	34.0	**71.1**	**60.4**	**35.3**	**39.4**

	R-Net				uniform R-Net				adversarial R-Net			
Task	P	V	NE	CN	P	V	NE	CN	P	V	NE	CN
Max	55.0	68.3	44.0	42.6	56.3	68.9	43.8	40.7	**60.0**	**70.0**	**44.5**	**42.9**

Table 6. *Accuracy (%) on the CBT dataset. In bold, the best result per architecture.*

Performance on the CBT dataset are displayed in Table 6. Because of the size of this dataset, we didn't repeat the training 10 times but only once. Results of the uniform training seem similar to the performance of the standard reader in this case but **the accuracy of the models trained with our adversarial protocol remains higher than others'**. This last experiment shows that augmenting the data distribution with random adversarial examples might not help the training as it was explained by Jia and Liang (2017). But we show that even in that case the protocol we propose is able to generate smart adversarial examples that will finally help the reader to improve its overall performance.

6. Visualizations and analysis

In this section, we present a series of analysis of the behavior of the competitive networks to better understand how the adversarial setting affects the training. We propose to analyze the probabilities of obfuscation of the different words of a given input (d, q, a).

In order to better understand how the obfuscation network learns from the reader behaviour during the adversarial protocol, Figure 5 depicts the rewards that the obfuscation network expects for each word of a document after several rounds of the game. Given a tuple (d, q) where d is a clear document and q a query, and assuming the document contains k words, we generate k corrupted documents where one word is obfuscated in each of them. We then feed the obfuscation network with these corrupted data and report the results. The expected rewards from the reader are displayed in green on the document. A strong intensity means that a high reward is expected.

Document:
A cheap Jamaican restaurant, please.
If you can not find one, then Mediterranean.
There are no Jamaican restaurants listed. I will look for Mediterranean.
Okay what is the phone number and postcode of that Mediterranean place?
The Gardenia is located in the centre at postcode C.B 2, 3 L.L and can be reached at 01223 356354. Can I get anything else for you?
Thank you.
Thank you for using the Cambridge restaurant system. Goodbye.

Question: What is the pricerange?

Document:
I am looking for a restaurant in the center of town that serves European food.
Eraina would fit the bill.
What are the address, phone number, and price range?
It's pretty expensive. It's at Free School Lane City Centre, and you can call them at 01223 368786. Anything else I can help you with?
No, that will be all. Thanks!
You're welcome!

Question: What is the area?

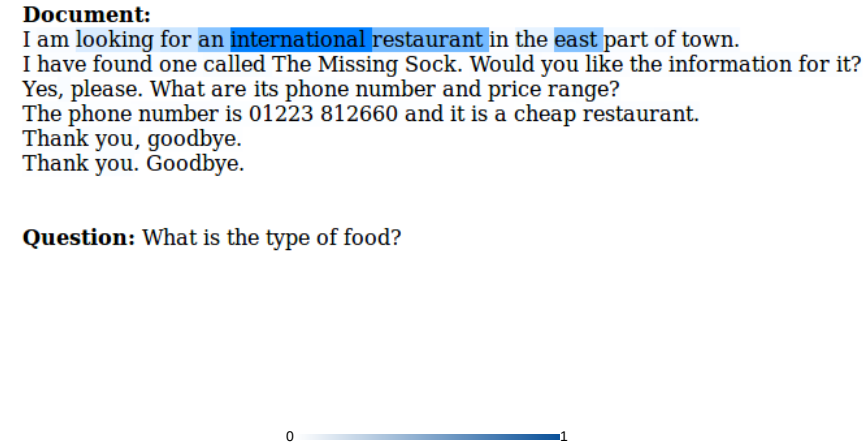


Figure 5. *Rewards expected by the obfuscation network after 100 rounds over a Cambridge dialog.*

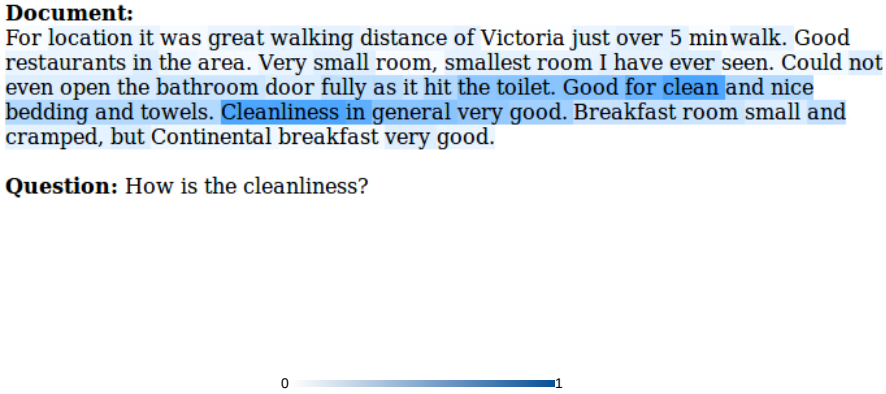


Figure 6. *Rewards expected by the obfuscation network after 100 rounds over a TripAdvisor review.*

We see that the obfuscation network tends to obfuscate some important keywords of the dialogs in Figure 5. Furthermore, the obfuscation network is not pointing on a single word but it points on a word and on its neighborhood. This could be a consequence of the encoding which is not only a representation of a single word but a representation of a word in its context. In Figure 6, we can see that the obfuscation network tends to affect a high probability of getting a reward for multiple words of the review. This can be a consequence of the performance of the reader on this dataset. Indeed if the reader is not generally confident about its answers, small changes in the reviews could lead to fool it. However, we can see on the figure that the most probable regions obfuscated by the obfuscation network refer to the cleanliness of the hotel which is coherent with the question.

7. Conclusion and future work

In this paper, we propose an adversarial learning protocol to train coupled deep neural networks for the task of machine reading. We propose two baselines, a Logistic Regression and an Attention-Sum Reader, on the three datasets used for our experiments. Then we experiment our adversarial learning protocol on two main types of neural architectures based on state-of-the-art machine reading models: a `GMemN2N` and a `R-Net`. In addition, we compare our adversarial protocol to a protocol based on a uniform corruption of data.

On all the reported experiments, the models trained with our novel protocol outperform the equivalent models trained with a standard supervised protocol or a protocol that introduces a uniform noise in the data which correspond to the more classic approach of dropout. Moreover, our adversarial protocol seems to improve the stability of the models' performance. Indeed, the variance of the results decreased when the training was done in an adversarial setup. We propose several visualizations that allow interpreting how the reader produces an answer, and which parts of the document are crucial for it to take its decision.

In future work, we plan to improve this novel protocol through an active question-answering task. Indeed the choice to only let the obfuscation network remove a single word might not be optimal. We would like to let it obfuscates multiple words while letting the reader the possibility to ask for revealing several words that might help it during training. Finally, we are currently investigating an adaptation of this protocol to Visual Question Answering.

The lack of robustness against adversarial examples and the difficulty to train deep neural networks with a limited set of data is not a specificity of language processing. This adversarial way of training deep neural networks should not be restricted to text documents but we think that it can also be useful in other domains.

8. References

Chen D., Bolton J., Manning C. D., "A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task", *CoRR*, 2016.

Cui Y., Chen Z., Wei S., Wang S., Liu T., Hu G., "Attention-over-Attention Neural Networks for Reading Comprehension", *ACL*, 2017.

Dauphin Y., Fan A., Auli M., Grangier D., "Language Modeling with Gated Convolutional Networks", *ICML*, 2017.

Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A. C., Bengio Y., "Generative Adversarial Nets", *NIPS*, 2014a.

Goodfellow I. J., Shlens J., Szegedy C., "Explaining and Harnessing Adversarial Examples", *CoRR*, 2014b.

Guo X., Klinger T., Rosenbaum C., Bigus J. P., Campbell M., Kawas B., Talamadupula K., Tesauro G., "Learning to Query, Reason, and Answer Questions On Ambiguous Texts", 2017.

Hermann K. M., Kociský T., Grefenstette E., Espeholt L., Kay W., Suleyman M., Blunsom P., "Teaching Machines to Read and Comprehend", *CoRR*, 2015.

Hill F., Bordes A., Chopra S., Weston J., "The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations", *CoRR*, 2015.

Jia R., Liang P., "Adversarial Examples for Evaluating Reading Comprehension Systems", *Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

Kadlec R., Schmid M., Bajgar O., Kleindienst J., "Text Understanding with the Attention-Sum Reader Network", *CoRR*, 2016.

Kim Y., "Convolutional Neural Networks for Sentence Classification", *EMNLP*, 2014.

Kingma D. P., Ba J., "Adam: A Method for Stochastic Optimization", *CoRR*, 2014.

Liu F., Perez J., "Dialog state tracking, a machine reading approach using Memory Network", *EACL*, 2017a.

Liu F., Perez J., "Gated End-to-End Memory Networks", *EACL*, 2017b.

Maaten L., Chen M., Tyree S., Weinberger K. Q., "Learning with Marginalized Corrupted Features", *in* S. Dasgupta, D. Mcallester (eds), *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, vol. 28, JMLR Workshop and Conference Proceedings, p. 410-418, 2013.

Miyato T., Dai A. M., Goodfellow I. J., "Virtual Adversarial Training for Semi-Supervised Text Classification", *CoRR*, 2016.

Nguyen T., Rosenberg M., Song X., Gao J., Tiwary S., Majumder R., Deng L., "MS MARCO: A Human Generated MAchine Reading COmprehension Dataset", *CoRR*, 2016.

Pennington J., Socher R., Manning C. D., "GloVe: Global Vectors for Word Representation", *Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532-1543, 2014.

Rajpurkar P., Zhang J., Lopyrev K., Liang P., "SQuAD: 100, 000+ Questions for Machine Comprehension of Text", *EMNLP*, 2016.

Richardson M., Burges C. J., Renshaw E., "MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text", *Proc. EMNLP*, 2013.

Rojas-Barahona L. M., Gasic M., Mrksic N., Su P.-H., Ultes S., Wen T.-H., Young S. J., Vandyke D., "A Network-based End-to-End Trainable Task-oriented Dialogue System", *EACL*, 2017.

Seo M. J., Kembhavi A., Farhadi A., Hajishirzi H., "Bidirectional Attention Flow for Machine Comprehension", *CoRR*, 2016.

Shen Y., Huang P.-S., Gao J., Chen W., "ReasoNet: Learning to Stop Reading in Machine Comprehension", *CoCo@NIPS*, 2016.

Silver D., Huang A., Maddison C. J., Guez A., Sifre L., van den Driessche G., Schrittwieser J., Antonoglou I., Panneershelvam V., Lanctot M., Dieleman S., Grewe D., Nham J., Kalchbrenner N., Sutskever I., Lillicrap T. P., Leach M., Kavukcuoglu K., Graepel T., Hassabis D., "Mastering the game of Go with deep neural networks and tree search", *Nature*, vol. 529 7587, p. 484-9, 2016.

Sordoni A., Bachman P., Bengio Y., "Iterative Alternating Neural Attention for Machine Reading", *CoRR*, 2016.

Srivastava N., Hinton G. E., Krizhevsky A., Sutskever I., Salakhutdinov R., "Dropout: a simple way to prevent neural networks from overfitting", *Journal of Machine Learning Research*, vol. 15, n° 1, p. 1929-1958, 2014.

Tang D., Qin B., Liu T., "Aspect Level Sentiment Classification with Deep Memory Network", *EMNLP*, 2016.

Tesauro G., "Temporal Difference Learning and TD-Gammon", *Commun. ACM*, vol. 38, n° 3, p. 58-68, March, 1995.

Trischler A., Wang T., Yuan X., Harris J., Sordoni A., Bachman P., Suleman K., "NewsQA: A Machine Comprehension Dataset", *Rep4NLP@ACL*, 2017.

Trischler A., Ye Z., Yuan X., Bachman P., Sordoni A., Suleman K., "Natural Language Comprehension with the EpiReader", *EMNLP*, 2016.

Wang H., Lu Y., Zhai C., "Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach", *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, ACM, New York, NY, USA, p. 783-792, 2010.

Wang T., Yuan X., Trischler A., "A Joint Model for Question Answering and Question Generation", *CoRR*, 2017a.

Wang W., Yang N., Wei F., Chang B., Zhou M., "Gated Self-Matching Networks for Reading Comprehension and Question Answering", *ACL*, 2017b.

Weston J., Bordes A., Chopra S., Mikolov T., "Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks", *CoRR*, 2015.

Weston J., Chopra S., Bordes A., "Memory Networks", *CoRR*, 2014.

Yu H., Munkhdalai T., "Neural Semantic Encoders", *Proceedings of the conference. Association for Computational Linguistics. Meeting*, vol. 1, p. 397-407, 2017.

Yuan X., Wang T., Gülçehre C., Sordoni A., Bachman P., Zhang S., Subramanian S., Trischler A., "Machine Comprehension by Text-to-Text Neural Question Generation", *Rep4NLP@ACL*, 2017.

Notes de lecture

Rubrique préparée par Denis Maurel

Université François Rabelais Tours, LI (Laboratoire d'informatique)

Mohamed Zakaria KURDI. Traitement automatique des langues et linguistique informatique 2. Sémantique, discours et applications. *Wiley-Iste*. 2018. 323 pages. ISBN 978-1-78405-185-3.

Lu par **Eleonora MARZI**

Université de Bologna (Italie)

Au cours de la dernière décennie, le progrès de certaines technologies – notamment l'augmentation de la puissance de calcul des machines – et une conjoncture historique particulière, qui voit la communication toujours plus orientée vers le multilinguisme et les grandes masses de données, font du TAL une discipline qui soulève de nouveaux défis. L'auteur, Mohamed Zakaria Kurdi, propose un aperçu approfondi des études existantes sur le langage à travers l'informatique, domaine interdisciplinaire par excellence. Avec une approche tant empirique que pratique, et en mettant « sur un pied d'égalité les modèles linguistiques et cognitifs, les algorithmes et les applications informatiques » ce texte parvient à donner un ample aperçu qui prend en considération les travaux classiques, mais aussi les plus contemporains.

Contenu et structure

Le livre se structure en quatre chapitres, répartis de manière équilibrée entre une approche théorique et une approche concrète. Le premier chapitre « *La sphère du lexique et des connaissances* » se concentre sur le lexique et la représentation des connaissances en introduisant la sémantique lexicale, en illustrant les bases de données lexicales et les ontologies. Le traitement des éléments de sémantique lexicale est approfondi tant au niveau de l'extension du sens, qu'au niveau de la pragmatique : Mohamed Zakaria Kurdi donne un aperçu des théories fondamentales du sens lexical, pour tout processus de catégorisation, en abordant l'approche aristotélicienne, l'approche sémique et componentielle du linguiste, Louis Hjelmslev, la sémantique du prototype de Eleanor Rosch et enfin la théorie du lexique génératif de James Pustejovsky. Quant à l'énumération de bases de données lexicales nous trouvons WordNet – avec une référence au passage à EuroWordNet, la base de noms propres Prolex, la base de données lexicales Brulex et la base Lexique. Le chapitre se conclut avec une section dédiée aux représentations formelles de la connaissance et aux ontologies, où sont illustrés les réseaux sémantiques de Quillian, les graphes conceptuels proposés par John Sowa, les

schémas de Marvin Minsky et les scripts de Schank et Abelson. En ce qui concerne les ontologies, certains projets assez représentatifs à l'heure actuelle y sont illustrés. On trouve DOLCE (*Descriptive Ontology for Linguistic and Cognitive Engineering*) et SUMO (*Suggested Upper Merged Ontology*).

Le deuxième chapitre « *La sphère de la sémantique* » propose deux approches dont l'auteur nous présente les évolutions. La première porte sur la sémantique combinatoire, qui inclut à son tour une variété d'approches comme la sémantique interprétative, la grammaire des cas et enfin la théorie sens-texte. La deuxième approche proposée est celle de la sémantique formelle qui s'est aujourd'hui assez répandue en raison du fait qu'elle se trouve à l'origine de toutes les applications dédiées au traitement computationnel du sens. À propos de cette deuxième approche, dont les fondements se trouvent dans le livre *Principia Mathematica* de Bertrand Russell et Alfred North Whitehead, l'auteur fournit les définitions de base des types de logiques les plus employées : la logique propositionnelle, la logique du premier ordre et celles d'ordre supérieur, la logique modale et la logique dynamique. Le chapitre se termine avec l'illustration du lambda calcul, théorisé en 1936 par le mathématicien Alonzo Church, et d'autres types de logiques comme celle d'ordre supérieur au premier ordre.

Le troisième chapitre s'ouvre à l'analyse du discours à travers l'apport de travaux qui vont de la linguistique à la critique littéraire, ce qui souligne encore une fois la perspective fort interdisciplinaire de cette œuvre. La première section est dédiée à l'illustration des notions de base (discours, parole, phrase, récit, texte), dans la seconde section sont présentés quelques domaines applicatifs considérés comme représentatifs, sans prétendre à l'exhaustivité, comme l'auteur le souligne sans cesse. L'illustration des notions de base se fait en référence à des travaux classiques : pour définir l'« énonciation », Mohamed Zakaria Kurdi s'appuie sur la subjectivité du langage d'Émile Benveniste, pour traiter les « déictiques » il se sert de réflexions de Roman Jakobson, et pour les « acteurs de la communication », il cite Ferdinand de Saussure. Sont également traités les notions de contexte, l'intertextualité et la transtextualité, la structuration du discours et les phénomènes de cohérence. La pragmatique occupe une section à part où l'on trouve les actes du langage de John Austin et John Searle. La seconde section du troisième chapitre est dédiée aux approches computationnelles du discours qui, comme les aspects théoriques qui les soutiennent, sont très divergentes. La variété des techniques de segmentation du discours est assez riche (base de n-grammes, réseaux bayésiens, analyse sémantique latente ou réseaux neuronaux), le cadre théorique présenté pour l'analyse automatique du discours est la théorie de la structure rhétorique, RST, (*Rhetorical Structure Theory*) et le cadre sémantique de référence est la théorie de représentation du discours, DRT, (*Discourse Representation Theory*). Le chapitre se

conclut par le traitement de l'anaphore qui occupe une section à part à cause de sa complexité.

Le quatrième chapitre « *La sphère des applications* » conclut l'aperçu en unifiant les notions théoriques linguistiques traitées tout au long de l'ouvrage avec d'autres sources de connaissances afin de construire des logiciels applicatifs appelés « linguiciels ». En particulier le chapitre aborde les linguiciels sous trois aspects : leur cycle de développement, leur architecture et leur évaluation. En ce qui concerne les architectures, on peut en imaginer plusieurs formes : la sélection de l'auteur se limite aux architectures les plus pertinentes pour le TAL. On traite les architectures sérielles, les architectures centrées sur les données, les architectures orientées objet et les architectures multi-agents. Les méthodes d'évaluation présentées sont le test structural (aussi appelé *whitebox test*), et l'évaluation de type quantitatif et qualitatif. Les applications traitées sont celles ayant un rapport avec la traduction automatique, dont l'auteur donne un aperçu historique de leurs débuts (1940) jusqu'à nos jours. Il donne aussi un aperçu des techniques employées, comme l'approche directe, l'approche par transfert, l'approche dite par pivot ou interlangue, l'approche à base d'exemple TBE et l'approche statistique. Une autre application traitée est la recherche d'information (RI) où sont énumérées les différentes approches utilisées aujourd'hui. Parmi les plus intéressantes on peut citer : les approches vectorielles et les approches fondées sur les groupements (*clustering*) qui sont également signalées par les dernières études sur le *deep learning* et sur l'intelligence artificielle. Ce trait de profonde actualité se reflète aussi à la fin du chapitre où on trouve une section dédiée au traitement des grandes masses de données et à l'extraction d'information, en particulier ayant trait à l'analyse des sentiments.

Commentaires

L'ouvrage se définit comme un aperçu de la discipline du traitement automatique des langues traitant d'une approche aussi bien théorique que pratique : les principales théories de la représentation de la connaissance et l'illustration des principales applications et bases de données existantes. L'interdisciplinarité du TAL se reflète dans l'ampleur des sujets traités dans l'ouvrage : la sémantique lexicale, la représentation des connaissances, l'analyse du discours et les applications s'entrecroisent avec des références aux travaux classiques et contemporains, ce qui montre la volonté de l'auteur de dresser aussi un aperçu chronologique.

Avec un agencement propre et méthodique, chaque chapitre est organisé de la même manière : une première partie, où sont illustrées les notions de base, est suivie par une seconde dans laquelle il est question des structures computationnelles, et des méthodes pour appliquer ces théories aux outils informatiques. Le style est clair et l'organisation du contenu aide le lecteur à suivre la variété des sujets qui vont de la

linguistique à l'intelligence artificielle, de l'informatique à la logique, de la morphologie aux transcriptions en codes.

L'ouvrage est complété par une bibliographie très exhaustive et détaillée qui tient compte de l'ampleur chronologique et thématique, et une assez claire structuration du sommaire compense le manque d'amplitude de l'index. Par rapport au paratexte, on signale la présence du sommaire du premier tome *Traitement automatique des langues et linguistique informatique 1*, qui est dédié aux notions fondamentales de la matière et qui confère à l'œuvre une complétude remarquable.

Tout aperçu porte en soi un trait spécifique, en couvrant un sujet large l'auteur est obligé de fournir des explications à propos de ses choix. Il déclare, à plusieurs occasions, son intention de passer en revue les plus importantes théories sans toutefois prétendre à l'exhaustivité. De fait, un choix s'impose, puisqu'il est impossible d'aborder la totalité des expériences existantes, le critère adopté sera celui du « plus représentatif » qui s'applique à un panorama francophone. L'ouvrage possède une perspective ample qui explique des concepts, tout en fournissant des suggestions et des sources bibliographiques pour approfondir davantage la recherche. L'ampleur des sujets traités fait ce cet ouvrage un outil précieux pour s'orienter dans l'évolution rapide et riche de la discipline du TAL.

François RASTIER. Faire sens. De la cognition à la culture. *Éditions Classiques Garnier*. 2018. 261 pages. ISBN 978-2-406-07413-7.

Lu par **Guy PERRIER**

Université de Lorraine – Loria

François Rastier propose une réorientation de la linguistique comme science sociale et de la culture. S'opposant à la conception logico-grammaticale du langage, il considère que le sens des textes n'est pas le résultat d'un calcul symbolique, mais le fruit d'une pratique interprétative sous forme d'un parcours de formes d'expression liées à des formes sémantiques. Et ce parcours met en jeu une dimension culturelle qui joue un rôle fondamental.

Pour qui veut découvrir l'œuvre de François Rastier, ce livre n'est pas le meilleur point d'entrée. J'en ai fait la douloureuse expérience. Étant totalement ignorant du contenu de ses travaux, je me suis dit, un peu naïvement, que la lecture du livre serait l'occasion de me familiariser avec eux. Or, l'univers de Rastier est peuplé d'une foule de notions étrangères au sens commun, et même à l'espace conceptuel de la plupart des chercheurs en TAL ; chaque page fait appel à ces notions qui ne sont pas redéfinies et qui nécessitent d'aller voir dans les écrits précédents de Rastier pour se les approprier. En plus, le livre est avare d'exemples qui pourraient nous en donner l'intuition. Toutefois, dans ma recherche, j'ai eu la

chance de tomber sur un texte de Philippe Gréa, « *La Perception sémantique* »[1], particulièrement pédagogique. Même si le sujet du texte ne recoupe pas complètement celui du livre de Rastier, il l'éclaire beaucoup et je voudrais lui rendre hommage. Je vous demande donc par avance de bien vouloir m'excuser de cette revue de néophyte, avec tous les manques qu'elle peut comporter. Je n'ai pas souhaité non plus être exhaustif ; le livre est très riche et je me suis arrêté sur quelques aspects qui intéressent plus particulièrement un chercheur dans le domaine du TAL.

Pourquoi ce livre ? Compte tenu des développements de ces dernières années en linguistique, Rastier a éprouvé le besoin d'actualiser ses propositions pour une réorientation de la linguistique comme science sociale et de la culture.

L'approche logico-grammaticale des langues

D'emblée, il oppose son approche de la linguistique à ce qu'il appelle l'approche logico-grammaticale, à laquelle il associe comme principaux artisans Chomsky, Fodor, Pylyshyn et Pinker. Selon cette approche, telle qu'elle est vue par Rastier, il n'y a pas de sémantique autonome des langues. Les langues sont un moyen d'accéder aux représentations du monde sous une forme logique. Dans cette conception, les signes linguistiques sont réduits à des symboles qui existent indépendamment les uns des autres avec une signification propre et immuable. Cette signification est aussi totalement séparée du symbole lui-même. Dans les textes, les symboles composent leurs significations selon le principe de compositionnalité à l'aide d'une grammaire universelle. L'interprétation d'un texte se réduit donc à un calcul sur des symboles, qui restent identiques à eux-mêmes durant tout le calcul. Cette conception donne le primat de la syntaxe sur la sémantique qui apparaît comme mécaniquement dépendante de la première. Elle est liée aussi à la conception du cerveau comme ordinateur dans sa fonction de cognition.

Rastier oppose à cela la conception saussurienne du signe linguistique dont les deux faces, le signifiant et le signifié, ne peuvent pas être séparées, le signifié n'étant pas extérieur à la langue. C'est pourquoi on peut dire que Saussure a été le premier à rendre possible une véritable sémantique linguistique.

Par ailleurs, le signe n'existe pas indépendamment du texte dans lequel il est présent. À l'opposé de l'approche logico-grammaticale, le local est déterminé par le global. Le sens d'un signe peut varier indéfiniment selon les occurrences. Il peut donc varier dans le temps et avoir une histoire.

La linguistique cognitive à la croisée des chemins

Si Rastier qualifie l'approche logico-grammaticale de cognitivisme orthodoxe, c'est parce que dans cette approche, il n'y a pas d'indépendance de la sémantique linguistique par rapport à l'univers conceptuel.

[1] https://halshs.archives-ouvertes.fr/halshs-01574243/document

La linguistique cognitive, reconnue comme telle, va plus loin en postulant que le langage n'a pas de spécificité par rapport à la cognition humaine et elle récuse toute sémantique logique en rapportant les phénomènes linguistiques à des processus mentaux. Elle a contribué à remettre en cause les postulats du cognitivisme orthodoxe. En particulier, la compréhension du sens y est plutôt vue comme une perception d'images mentales.

Rastier rapproche ces propositions des grammaires de construction, qui visent aussi à répondre aux faiblesses de l'approche chomskyenne. Dans les grammaires de construction, le rapport entre expression et sens est plus complexe, puisqu'il n'est pas défini seulement par un lexique, mais aussi par des constructions. Cependant, comme la linguistique cognitive, les grammaires de construction n'ont pas réussi à se dégager complètement du paradigme logico-grammatical : elles ne distinguent pas le signifié du concept et le sens est construit de bas en haut par composition d'unités élémentaires, alors que pour Rastier, c'est le global, le texte, qui détermine le local, le mot.

L'interprétation comme perception sémantique

Rastier, dans sa conception de l'interprétation sémantique d'un texte, s'appuie sur une relecture de Saussure. Selon lui, la présentation des idées de Saussure a été souvent tronquée : sa conception du signe linguistique ne se réduit pas à la dualité signifiant-signifié, mais le signe se définit aussi dans sa relation avec le contexte. Cela interdit de concevoir l'interprétation d'une expression comme un calcul permettant de composer le sens des sous-expressions.

Selon Rastier, le sens d'un texte est le fruit d'une perception plutôt que d'un calcul de représentations. Les unités sémantiques ne s'expriment pas comme unités discrètes relativement figées, mais comme des formes qui se profilent sur des fonds. Le sens d'un texte résulte du parcours de ces formes sémantiques. Ce parcours est évolutif, les formes se dissolvant dans les fonds par diffusion ou émergeant des fonds par sommation.

De la communication à la transmission

Après avoir abordé la question du langage sous l'angle de la cognition, Rastier montre comment on retrouve ses idées quand on l'aborde sous l'angle de la communication. La communication langagière, que Rastier appelle transmission, ne se réduit pas à un processus de codage-décodage, tel que le schéma standard de la communication le décrit. Ce schéma est parfaitement symétrique. Or, le message langagier n'est pas perçu de la même façon par l'émetteur et le récepteur.

Pour Rastier, une transmission est un fait culturel. Elle met en jeu non seulement deux protagonistes, mais aussi tout un contexte culturel. La transmission est fondamentalement une transmission culturelle, vue non pas comme un processus déterministe, mais comme une réappropriation active.

En conclusion, j'espère que ces quelques commentaires sur le livre de Rastier donneront envie à certains d'aller plus loin en se plongeant eux-mêmes dans sa lecture. Nous, chercheurs en TAL, aurions intérêt à prêter davantage attention aux

idées de Rastier, dans un souci de coller davantage à la réalité complexe des langues. La linguistique de corpus, par le rôle essentiel qu'elle attribue au contexte, parce qu'elle permet de lier expression et contenu et de prendre en compte la détermination du local par le global, est un champ d'application naturel des idées de Rastier. L'apprentissage automatique statistique est un des moyens pour le TAL d'investiguer sous cet angle la linguistique de corpus.

Maintenant, faut-il jeter à la poubelle l'approche logico-grammaticale, sur laquelle sont fondés l'essentiel de la recherche en TAL et tous ses acquis ? Faut-il construire un modèle formel alternatif fondé sur les idées de Rastier, sachant que leur très grande complexité est un obstacle à leur formalisation ? Ou faut-il enfin trouver une synthèse entre deux approches apparemment inconciliables ? L'avenir permettra de dire si le TAL a su s'approprier les idées de Rastier.