Acta Cybernetica **23** (2018) 801–813.

ℓ_1 Regularization of Word Embeddings for Multi-Word Expression Identification

Gábor Berend[a]

Abstract

In this paper we compare the effects of applying various state-of-the-art word representation strategies in the task of multi-word expression (MWE) identification. In particular, we analyze the strengths and weaknesses of the usage of ℓ_1-regularized sparse word embeddings for identifying MWEs. Our earlier study demonstrated the effectiveness of regularized word embeddings in other sequence labeling tasks, i.e. part-of-speech tagging and named entity recognition, but it has not yet been rigorously evaluated for the identification of MWEs yet.

Keywords: sparse coding, multi-word expressions, word embeddings

1 Introduction

Multi-word expressions (MWEs) are semantically coherent linguistic constructions including whitespace characters like "*paternal leave*" and "*shut off*" [11, 22]. The identification and proper treatment of such expressions is an important and challenging task which can improve the performance of various natural language processing (NLP) applications such as the extraction of opinionated expressions [2] and machine translation [6].

Continuous word embeddings have become prevalent in a variety of NLP tools due to their intriguing property of being able to capture both semantic and syntactic properties of word forms [15]. Such dense word representations have been successfully applied in many NLP analyzers such as syntactic parsers and part-of-speech taggers [8, 19, 20].

Instead of the typical approach of regarding the dense vectorial representations of words as the discriminative features, here we investigate the utilization of ℓ_1 regularized sparse word embeddings, which has been shown to provide substantial gains in the tasks of part-of-speech (POS) tagging and named entity recognition (NER) [3] over multiple languages. Besides utilizing regularized word embeddings we do not rely on any other (linguistic) resources in order to keep the proposed approach easily applicable to new languages.

[a]Department of Informatics, University of Szeged, E-mail: berendg@inf.u-szeged.hu

DOI: 10.14232/actacyb.23.3.2018.5

2 Previous work

In [29], the authors use a sequence labeling framework for the detection of a special kind of MWE, namely light verb constructions. In [23], a wider range of MWEs are studied by applying a standard chunking representation and proposing a feature-rich discriminative sequence tagging algorithm for the proposed problem. The feature-rich representation of typical approaches often assume the existence of additional linguistic resources, such as gazetteers containing highly indicative words for certain kinds of MWEs, part-of-speech taggers and even syntactic parsers [27]. While the use of such external resources is legitimate from a linguistic perspective, it makes these approaches less robust for utilizing them in languages where such resources do not exist.

Word embeddings, however, are capable of representing the syntactic and/or semantic nature of word forms and can be trained in an unsupervised manner [15]. For this reason, sequence labeling models which rely on word embeddings can implicitly incorporate syntactic/semantic knowledge without an explicit reliance of NLP parsers. Word embeddings have became commonly used in many MWE-related tasks due to their intriguing properties. For instance, word embeddings are user in order to improve the quality of the translation of phrasal verbs in [7].

The authors of [20] contrast the effects of utilizing differently trained dense word embeddings and Brown clustering for the application of classical uni-and bigram-based models in MWE identification besides part-of-speech tagging, syntactic chunking and named entity recognition. They found that models which had access to word embeddings had a consistent advantage over models which classified tokens based on unigram features. At the same time they report that there was no word embedding approach that would have a clear advantage over the others for all the sequence labeling tasks and that one can perform competitively with models that rely on continuous word embeddings for certain sequence labeling tasks by relying on Brown cluster identifiers of word forms.

Our earlier work has demonstrated that substantial improvements can be gained in the tasks of part-of-speech tagging and named entity recognition if the discriminative features that are used by the sequence classifiers are derived from the ℓ_1 regularized variants of dense word representations instead of the dense vectorial representation of word forms [3]. In this study, we investigate and rigorously compare the applicability of this approach for the task of MWE identification.

Unsupervised word clusters (e.g. in the form of Brown clustering [5]) have also been frequently employed for representing words in various sequence labeling tasks for NER [21, 9], chunking [26], POS tagging [24] and MWE identification [23].

MWEs are in the focus of multiple other research efforts. The approach presented in [4] is among the alternatives for acquiring multiword lexicons in an unsupervised manner using n-gram lattices.

3 Experimental settings

The experimental setting in this study extends that of [3], where we showed that sequence models relying on features derived from the ℓ_1 regularized versions of dense word embeddings perform competitively or even better than classical models for part-of-speech tagging and named entity recognition. We released the code base used in our experiments at `https://github.com/begab/tacl_sparse_coding`.

3.1 Applying ℓ_1 regularized word embeddings

The approach described in [3] relies on continuous word embeddings, such as word2vec [15] and Glove [18]. Word embeddings map the symbolic elements of the vocabulary of some language to m-dimensional real-valued vectors ($\mathbf{x} \in \mathbf{R}^m$) such that syntactically and/or semantically similar word forms get assigned vectors which point in similar directions. For a vocabulary consisting of n distinct word forms, these word embeddings can be stacked to form a $X \in \mathbf{R}^{m \times n}$ matrix. Such word embeddings can be constructed with a variety of open-source tools[1][2] and require no resources other than raw, unannotated text corpora for which reason their usage has become ubiquitous in many NLP applications.

The ℓ_1 regularization of word embeddings takes place using dictionary learning [14], which decomposes the original embedding matrix X by solving the following optimization problem

$$\min_{D \in \mathcal{C}, \alpha} \|X - D\alpha\|_F^2 + \lambda \|\alpha\|_1, \tag{1}$$

in which $\mathcal{C}$ is the convex set of matrices of column vectors having an ℓ_2 norm of at most one, matrix $D \in \mathbf{R}^{m \times k}$ acts as the shared dictionary across the word embedding signals, and the columns of the sparse matrix $\alpha \in \mathbf{R}^{k \times n}$ contain the coefficients for the linear combinations of each of the n observed signals.

Dictionary learning has two parameters, namely k, which is the number of basis vectors to be included in the dictionary matrix D, and the regularization coefficient λ, which implicitly controls the amount of non-zero coefficients in α; that is, the amount of basis vectors utilized in the reconstruction of input word embeddings. Assuming that the vectorial representation of some word form x is located in the i^{th} column of the embedding matrix X, sparse discriminative features are derived from those positions of the i^{th} column of α that contain non-zero coefficients. In the remainder of the paper, we shall refer to sequence classifiers which assign discriminative features to word forms this way as sparse models.

In contrast to sparse models, scalars comprising the original dense vectors assigned to word forms can act as discriminative features as well. This means that each token is described by m scalars, whereas in the sparse scenario tokens are described by a fraction of indicator variables depending on the regularization parameter chosen. We shall refer to sequence classifiers that treat word forms this way as dense models.

[1] `https://code.google.com/archive/p/word2vec/`
[2] `http://nlp.stanford.edu/projects/glove/`

3.2 The dataset

The dataset that we conducted our experiments on is the WIKI50 corpus [28]. WIKI50 is a collection of 50 Wikipedia articles in which all the occurrences of 4+6 different kinds of multi-word units have been annotated manually. Proper nouns often consist of multiple tokens, which is why the dataset contains annotations for the 4 standard named entity (NE) categories, i.e. *Organization*, *Person*, *Location* and *Miscellaneous*. The dataset also distinguishes the following MWEs (with examples in parenthesis): *Noun Compounds* ("public transportation"), *Adjectival Compounds* ("monkey styled"), *Verb-Particle Constructions* ("went on"), *Light-Verb Constructions* ("opens fire"), *Idioms* ("caught the eye of") and *Other* ("alter ego"). The dataset consists of 114,284 tokens and 4366 sentences originating from 50 Wikipedia articles.

When reporting detailed results for the individual MWE classes we focus on 8 different types of MWEs, as opposed to the 10 total classes distinguished in the WIKI50 corpus. This is due to the fact that we do not report results for the MWE categories *Idiom* and *Other* due to their highly infrequent nature. The above–mentioned categories have 19 and 21 occurrences over the entire WIKI50 dataset, respectively, meaning that more than half of the Wikipedia articles do not contain a single instance of these categories. The overall classification metrics that we report, however, do incorporate results on these two categories as well.

4 Experimental results

We use the CRFsuite [16] package to train first-order conditional random fields (CRF) [12] models as sequence classifiers. Unless otherwise stated, words at a certain position within a sequence are described by the (sparse or dense) features representing the given word and also those of its immediate neighbors. Features also incorporate relative token positions (whether a certain feature comes from the previous, actual or the successive token) that were taken into consideration. This means that for the dense model each token position is described by a vector in $\mathbf{R}^{3m}$.

The performance of the models we experiment with is evaluated using 50-fold cross-validation. Here we train 50 models, that is for making predictions for a Wikipedia article taken from the WIKI50 corpus we train one model based on the labeled token sequences of all the remaining 49 Wikipedia articles from the dataset. This way when making predictions about the tokens of a particular Wikipedia article, we can ensure that none of the sentences from the same Wikipedia article is used during the parameter estimation of the model making predictions for the given document.

For evaluation purposes, we used the same script that was released as part of the 2002 CoNLL shared task on named entity recognition [25]. Even though the script was released for a shared task on NER, it seamlessly adapts for any set of class labels. It provides precision, recall and F-score metrics for the individual MWE types and also for the entire sequence labeling task.

4.1 Comparing sparse and dense embedding-based models

In this section, we investigate the effects of deriving features from dense versus sparse word embeddings for the sequence classification model. We experimented with four popular continuous word embeddings, i.e. glove [18], polyglot [1], skip-gram (sg) and continuous bag-of-words (cbow) [15]. As for the polyglot embeddings we use the publicly available [3] 64-dimensional pre-trained embeddings, which are trained over an English Wikipedia dump also made accessible by the authors of [1]. In order to be able to objectively assess the quality of word embedding techniques it is vital that the embeddings should be trained under as similar circumstances as possible. For this reason we trained our own sg, cbow and glove embedding over the same corpus that is used for training polyglot embeddings.

When deriving sparse word representations like that described in Section 3.1, we set k, the number of basis vectors in the dictionary matrix D, to 1000 and choose the value for λ from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Depending on the value for λ, we found 0.5% to 5% of the coefficients in α to be non-zero, which means that the average number of features per word forms is between 5 (for $\lambda = 0.5$) and 50 (for $\lambda = 0.1$).

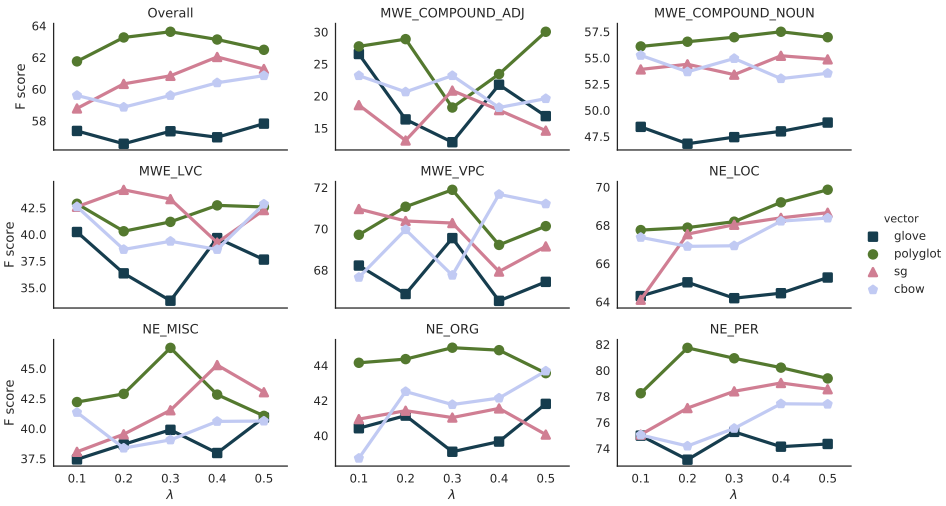

Figure 1: Overall results and a per-multi-word unit category breakdown of the F-scores as a function of the regularization parameter λ and the pre-trained word embedding algorithms.

Figure 1 contains results for the overall classification performance and its breakdown according to the different MWE classes (excluding *Idioms* and *Other* class, as discussed in Section 3.2). This table tells us that the overall performance peaks for polyglot word embeddings with a regularization coefficient of 0.3. This choice of the regularization parameter provides not only the best overall F-scores, but it

[3] `https://sites.google.com/site/rmyeid/projects/polyglot`

produces the best performance for multiple individual MWE types. The class of compound adjectives behaves in the least predictable way when altering the regularization coefficient λ. This is due to the fact that this MWE type is one of the least frequent classes, for which reason the misclassification of a few instances can have a dramatic effect overall. Increasing λ beyond a certain value (typically 0.3) has a detrimental effect on the performance for nearly all of the MWE types. The location NE category is a notable exception to this, as the identification performance for this category does not seem to degrade even for the highest level of regularization employed. Based on the entire contents of Figure 1, the regularization coefficient was set to 0.3.

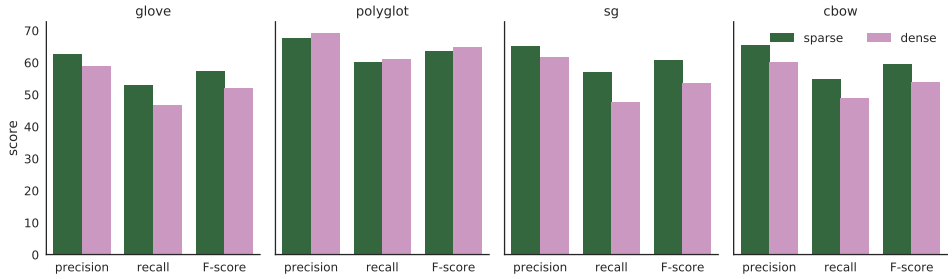

Figure 2: Overall results of models that utilize dense and sparse word embeddings-based features.

Figure 2 indicates that for most of the embedding types the features derived from sparse embeddings have a substantially better overall performance. The only exception is for polyglot embeddings where the sparse versions do not provide better results compared to the sequence classifier deriving features based on dense vectors. Figure 2 also indicates that polyglot embeddings obtained the best results for the task of MWE identification, hence comparative results in the remainder of the paper will be based on them.

4.2 Alternative models

In order to compare our word embedding-based results, we provide a variety of alternative approaches that will be presented and assessed next.

Feature-rich representation As an alternative to word embedding-based models, we evaluate a sequence classification model using a standard inventory of surface form features derived from the word identities themselves. The pool of feature templates is inspired by those made publicly available as part of CRFsuite [16]. We will use all the feature templates[4] included in the CRFSuite library, which derive features from word forms themselves but we do not include those features which are based on characters and character sequences comprising a word form. We omitted

[4] `https://github.com/chokkan/crfsuite/blob/master/example/pos.py`

character-based features as our primarily intention here is to compare the effects of word forms-derived features on sequence classification. The set of feature templates is listed in Table 1. Due to the high number of features induced by the templates, we shall refer to the models relying on them as feature rich (FR) models.

Feature template	
w_{t+j}	$-2 \leq j \leq 2$
$w_t \oplus w_{t+j}$	$1 \leq j \leq 9$
$w_t \oplus w_{t-j}$	$1 \leq j \leq 9$
$\oplus_{i=t+j}^{t+j+1} w_i$	$-2 \leq j \leq 1$
$\oplus_{i=t+j}^{t+j+2} w_i$	$-2 \leq j \leq 0$
$\oplus_{i=t+j-1}^{t+j+2} w_i$	$-1 \leq j \leq 0$
$\oplus_{i=t-2}^{t+2} w_i$	

Table 1: Feature templates applied by our feature-rich baselines for some target word w_t at position t within a sequence. $\oplus$ is a binary operator forming a feature from words and their relative positions within the sequence by concatenating them.

Brown clustering Brown clustering [5] is widely used to provide useful word representation in many NLP sequence labeling tasks [17, 9, 21, 24]. For this reason we also train a sequence classifier for identifying MWEs which represents word forms based on their Brown cluster identifier.

In our experiments, we used the implementation of [13] to perform Brown clustering[5]. The same Wikipedia articles which serve as input for learning word embeddings are employed for determining 1024 Brown clusters over the vocabulary. The word features that we derive from the Brown cluster identifiers of word forms are the $\{4, 6, 10, 20\}$-long prefixes of Brown cluster identifiers of the word forms.

Long-short term memory (LSTM) networks LSTMs [10] are an extension of recurrent neural networks (RNN), which provide a remedy for the vanishing/exploding gradient problem during backpropagation in RNNs via gating mechanisms. LSTMs are regarded as the state-of-the-art models for many sequence labeling tasks in NLP.

The authors of [19] released their bidirectional LSTM implementation for part-of-speech tagging[6]. We adapted their implementation for training bi-LSTM sequence classifiers to identify MWEs. We made two modifications to their default settings, i.e. we used word embedding features only (whereas [19] defines character-level embeddings as well) and we trained the model for 15 epochs (instead of 30). The reason why we did not employ character-level embeddings in our model was that we wanted to compare the effects of various word representations alone and not to conflate it with the joint usage of additional features. What is more, the

[5] `https://github.com/percyliang/brown-cluster`
[6] `https://github.com/bplank/bilstm-aux`

use of character-level embeddings would make the training procedure substantially slower (especially that we performed 50-fold cross–validation).

We initialize the word embeddings of the bi-LSTM model with polyglot embeddings; however, they were treated as the parameters of the model, meaning that they were updated during training. The pre-initialization step of the word embedding parameters of the model is essential for good performance as the Wiki50 corpus is too small to learn reliable word embeddings based on it alone from a randomly initialized state. We observed that evaluation metrics of the bi-LSTM model degrade substantially if pre-initialization is not applied.

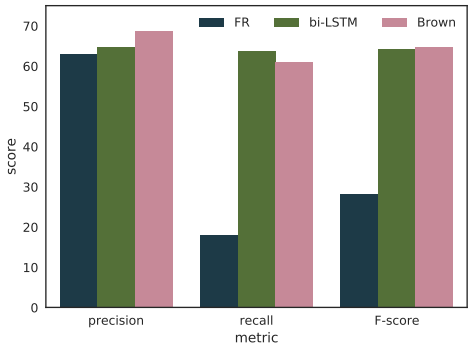

Figure 3: Overall results of the alternative models.

Figure 3 shows the overall results for the alternative models introduced previously. We observe that the feature–rich model performs the poorest mostly due to its low recall score. Another observation is that even though bi-LSTMs are considered as state-of-the-art approaches for sequence classification, it slightly underperforms the Brown clustering-based model, i.e. the bi-LSTM has an overall F-score of 64.22 as opposed to 64.90.

It should be mentioned that we managed to improve the scores of the bi-LSTM model by incorporating not only word embeddings, but character-level embeddings as well. Extending the model this way resulted in an overall F-score of 66.48 at the expense of a much longer training time. Furthermore, when we investigated the MWE-type specific changes in the scores, we realized that the overall improvement was due to improvements just for the named entity categories, whereas its ability to detect other types of non-NE MWEs either remained the same or even degraded slightly.

4.3 Detailed comparative results of different models

In order to gain a better insight into the performance of various models using conceptually different feature representations, we shall provide an MWE type-specific breakdown of the overall results. Table 2 provides an overview of the different models we investigated in our experiments. Inspecting Table 2, we see that precision

values tend to be higher than the recall scores with all the approaches and the bi-LSTM model seems to be the most balanced with respect to the gap between precision and recall scores.

method	Precision	Recall	F-score
polyglot sparse ($\lambda = 0.3$)	67.65	60.03	63.61
bi-LSTM	64.81	63.64	64.22
Brown	68.79	60.90	64.60
polyglot dense	69.24	61.07	64.90

Table 2: Comparison of the overall performance of conceptually different models.

Figure 4 includes the MWE-type specific breakdown of the individual models, which confirm that overall precision values tend to be higher compared to the recall scores. The only notable exception is the performance of the bi-LSTM model on the compound nouns, for which the precision scores are markedly lower compared to recall. This is due to the fact that the bi-LSTM model has a better ability to predict that category.

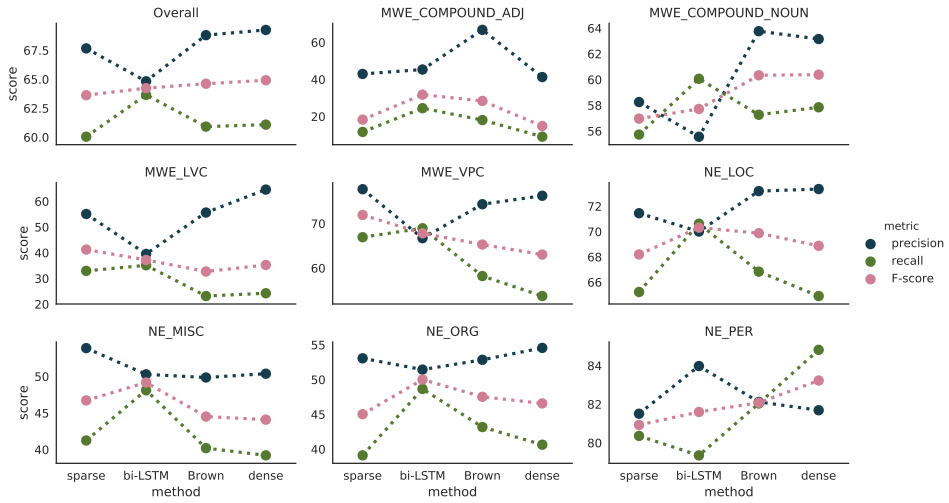

Figure 4: A per-MWE type comparison of the best performing models.

Figure 4 also elucidates the balanced nature of the bi-LSTM model in terms of the difference in precision and recall scores. The only exception to this balanced performance is the person NE type, for which it produces the highest precision–recall gap as the bi-LSTM model is less capable of predicting that particular category.

Looking at Figure 4 further, we can identify certain MWE categories for which certain approaches perform much better than others. The bi-LSTM has the best performance for named entity types apart from the person (NE_PER) category, the

Brown and dense models perform better than the other approaches for the compound noun category, whereas the sparse model achieves the best scores for the identification of light verb constructions (LVC) and verb-particle constructions (VPC).

5 Conclusions

In this paper, we investigated the applicability of sparse coding derived word features for the extraction of MWEs. Our experimental results demonstrate that the integration of sparse word features into sequence classifiers gives a performance competitive with state-of-the-art models, including bi-directional LSTMs. We should mention that the models applied here did not rely on POS taggers, syntactic parsers or gazetteers, implying that they can be conveniently adapted for the identification of MWEs in multiple languages without the need for any additional linguistic resources. Lastly, we found that sparse word representations seem to be the most suitable for the identification of verb-particle constructions and light verb constructions.

Acknowledgement

We gratefully acknowledge the support of the NVIDIA Corporation with the donation of the Titan X Pascal GPU used in this research.

References

[1] Al-Rfou, Rami, Perozzi, Bryan, and Skiena, Steven. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192. Association for Computational Linguistics, 2013.

[2] Berend, Gábor. Opinion expression mining by exploiting keyphrase extraction. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 1162–1170, 2011.

[3] Berend, Gábor. Sparse coding of neural word embeddings for multilingual sequence labeling. *Transactions of the Association for Computational Linguistics*, 5:247–261, 2017.

[4] Brooke, Julian, Snajder, Jan, and Baldwin, Timothy. Unsupervised acquisition of comprehensive multiword lexicons using competition in an n-gram lattice. *Transactions of the Association for Computational Linguistics*, 5:455–470, 2017.

[5] Brown, Peter F., deSouza, Peter V., Mercer, Robert L., Pietra, Vincent J. Della, and Lai, Jenifer C. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[6] Carpuat, Marine and Diab, Mona. Task-based evaluation of multiword expressions: A pilot study in statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 242–245, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[7] Cholakov, Kostadin and Kordoni, Valia. Using word embeddings for improving statistical machine translation of phrasal verbs. In *Proceedings of the 12th Workshop on Multiword Expressions, MWE@ACL 2016, Berlin, Germany, August 11, 2016.*, 2016.

[8] Collobert, Ronan, Weston, Jason, Bottou, Léon, Karlen, Michael, Kavukcuoglu, Koray, and Kuksa, Pavel. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[9] Derczynski, Leon, Chester, Sean, and Bøgh, Kenneth. Tune your brown clustering, please. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 110–117. INCOMA Ltd. Shoumen, Bulgaria, 2015.

[10] Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[11] Kim, Su Nam. *Statistical Modeling of Multiword Expressions.* PhD thesis, University of Melbourne, Melbourne, 2008.

[12] Lafferty, John D., McCallum, Andrew, and Pereira, Fernando C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289. Morgan Kaufmann Publishers Inc., 2001.

[13] Liang, Percy. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology, 2005.

[14] Mairal, Julien, Bach, Francis, Ponce, Jean, and Sapiro, Guillermo. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696. Association for Computing Machinery, 2009.

[15] Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[16] Okazaki, Naoaki. CRFsuite: a fast implementation of Conditional Random Fields (CRFs), 2007.

[17] Owoputi, Olutobi, Dyer, Chris, Gimpel, Kevin, Schneider, Nathan, and Smith, Noah A. Improved part-of-speech tagging for online conversational text with word clusters. In *In Proceedings of NAACL*, 2013.

[18] Pennington, Jeffrey, Socher, Richard, and Manning, Christopher. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, 2014.

[19] Plank, Barbara, Søgaard, Anders, and Goldberg, Yoav. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418. Association for Computational Linguistics, 2016.

[20] Qu, Lizhen, Ferraro, Gabriela, Zhou, Liyuan, Hou, Weiwei, Schneider, Nathan, and Baldwin, Timothy. Big data small data, in domain out-of domain, known word unknown word: The impact of word representations on sequence labelling tasks. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 83–93. Association for Computational Linguistics, 2015.

[21] Ratinov, Lev and Roth, Dan. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155. Association for Computational Linguistics, 2009.

[22] Sag, Ivan A., Baldwin, Timothy, Bond, Francis, Copestake, Ann, and Flickinger, Dan. Multiword Expressions: A Pain in the Neck for NLP. In *Proceedings of CICLing-2002*, pages 1–15, Mexico City, Mexico, 2002.

[23] Schneider, Nathan, Danchik, Emily, Dyer, Chris, and Smith, Noah A. Discriminative lexical semantic segmentation with gaps: Running the MWE gamut. *Transactions of the Association of Computational Linguistics*, 2:193–206, 2014.

[24] Stratos, Karl and Collins, Michael. Simple semi-supervised POS tagging. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 79–87. Association for Computational Linguistics, 2015.

[25] Tjong Kim Sang, Erik F. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158, 2002.

[26] Turian, Joseph, Ratinov, Lev, and Bengio, Yoshua. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394. Association for Computational Linguistics, 2010.

[27] Vincze, Veronika, Nagy, T. István, and Berend, Gábor. Detecting noun compounds and light verb constructions: A contrastive study. In *Proceedings of the Workshop on Multiword Expressions: From Parsing and Generation to the Real World*, MWE '11, pages 116–121, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[28] Vincze, Veronika, Nagy T., István, and Berend, Gábor. Multiword expressions and named entities in the Wiki50 corpus. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 289–295, Hissar, Bulgaria, September 2011. RANLP 2011 Organising Committee.

[29] Vincze, Veronika, T., István Nagy, and Zsibrita, János. Learning to detect English and Hungarian light verb constructions. *TSLP*, 10(2):6:1–6:25, 2013.

Received 28th May 2018