What's in an Embedding? Analyzing Word Embeddings through Multilingual Evaluation

Arne Köhn
Department of Informatics
Universität Hamburg
koehn@informatik.uni-hamburg.de

Abstract

In the last two years, there has been a surge of word embedding algorithms and research on them. However, evaluation has mostly been carried out on a narrow set of tasks, mainly word similarity/relatedness and word relation similarity and on a single language, namely English.

We propose an approach to evaluate embeddings on a variety of languages that also yields insights into the structure of the embedding space by investigating how well word embeddings cluster along different syntactic features.

We show that all embedding approaches behave similarly in this task, with dependency-based embeddings performing best. This effect is even more pronounced when generating low dimensional embeddings.

1 Introduction

Word embeddings map words into a vector space, allowing to reason about words in this space. They have been shown to be beneficial for several tasks such as machine translation (Botha and Blunsom, 2014), parsing (Lei et al., 2014), and named entity recognition (Passos et al., 2014). Recently, word embedding techniques have been studied for their mathematical properties (Levy and Goldberg, 2014b; Stratos et al., 2015), yielding a better understanding of the underlying optimization criteria. However, word embeddings have mostly been studied and evaluated on a single language (English). Therefore, validation on languages other than English is lacking and the question whether word embeddings work the same way across languages has not been empirically evaluated. Evaluations of complex systems – such as parsers – employing word embeddings generally give only little insight into the type of contribution to the result and the structure of word embeddings.

We aim to fill these gaps by evaluating several word embedding algorithms on a set of different languages using tasks that enable additional insight into the learned structures using easily obtainable data. At the same time, we provide baseline results for using word embeddings in several syntax-based classification tasks.

We focus on syntax-related measures because data is available for several languages and we expect a correlation with usefulness of word embeddings for syntax-related tasks such as named entity recognition, parsing, and morphological analysis.

2 Related Work

Previous approaches to word embedding evaluation have either used relatively basic word finding and classification tasks (as this paper also proposes) or application-oriented end-to-end evaluations as part of a larger system. Word finding tasks are of the form "given a pair of words (x, y), find a y' for a given x'", e.g. given (Rome, Italy), find a word for Oslo. These tasks have been introduced by Mikolov et al. (2013a). The downside of this kind of task is that the data is not readily available and has to be constructed for each language. This type of evaluation primarily describes the similarity between vector differences and not similarity between vectors. In addition, Levy et al. (2015) showed for this task that word embedding-based classifiers actually mostly learn whether a word is a *general hypernym* and not, as would be expected, the relation between two words.

Another approach to evaluate embeddings, used by Pennington et al. (2014) amongst others, is to rank a fixed set of words relative to a reference word. The results are then compared to human judgments, e.g. from the WS353 corpus (Finkelstein et al., 2002). This approach has a limited coverage and additional data is expensive to obtain.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2067–2073,
Lisbon, Portugal, 17-21 September 2015.

Botha and Blunsom (2014) propose to factorize word vectors into morpheme vectors to better capture similarities between morphologically related words and evaluate their word representations using log-bilinear language models based on their word vectors.[1] They measure model perplexity reduction relative to n-gram language models and include their model into a machine translation system, gaining between 0 (English → German) and 1.2 (English → Russian) BLEU points.

Lei et al. (2014) introduce a syntactic dependency parser using (amongst others) a low-rank tensor component for scoring dependency edges. This scoring can employ word embeddings. Doing so yields an improvement of 0.2 to 0.5 percentage points. If no Part-of-Speech (PoS) tags are available, this difference rises to up to four percentage points. Köhn et al. (2014) show that this gain from using word embeddings is even more pronounced in complete absence of morphological information (including PoS tags), reporting a difference of five to seven percentage points,depending on the language, using the same parser. With these findings, it can be assumed that word embeddings encode some kind of morphological information. Neither, however, investigated what kind of information the word embeddings actually contain.

3 The Embedding Algorithms

To assess the differences between embedding algorithms, we will evaluate six different approaches. The continuous bag-of-words (**cbow**) approach described by Mikolov et al. (2013a) is learned by predicting the word vector based on the context vectors. The skip-gram approach (**skip**) from the same authors is doing the reverse: it predicts the context word vectors based on the embedding of the current word. We use the version of **cbow** and **skip** as described in (Mikolov et al., 2013b) which use negative sampling, i.e. they train by distinguishing the correct word in its context against words not occurring in that context.

Levy and Goldberg (2014a) alter the skipgram approach by not using the neighboring words wrt. the sentence's word sequence but wrt. the dependency tree of the sentence. Therefore, the context of w is defined as all words that are either the head or dependents of w. We will call this approach **dep**.

[1] Their approach has not been evaluated in this paper as the corresponding code is not available as of now.

GloVe, introduced by Pennington et al. (2014), optimizes the ratio of co-occurrence probabilities instead of the co-occurrence probabilities themselves, getting rid of the negative sampling used for the approaches previously mentioned.

Stratos et al. (2015) describe a method to derive word embeddings using canonical correlation analysis. We will call this approach **cca**.

brown clusters (Brown et al., 1992) are constructed by clustering words hierarchically into a binary search tree in a way that maximizes mutual information for a language model. To construct an embedding for a cluster c, we use the following procedure: For each edge on the path from the root to c, add either 1 or -1, depending on the direction of descent. Because not every path has the same depth, we pad missing dimensions with 0. This way, we obtain an embedding interpretation of the clusters. Note that, in contrast to clustering embeddings, no information is lost.

4 Our Evaluation by Classification Tasks

We classify words separately according to several tasks with an L2-regularized linear classifier. All classification tasks are based on the word embedding of a single word alone, without any other information about the word or its context; in particular, the word's lexicalization is not used as a feature. By using the continuous features directly instead of clustering them (as e.g. done by Bansal et al. (2014)), we ensure that no information is lost during preprocessing.

All tasks can be carried out on dependency treebanks with morphological annotation. From each word in the treebank, we extract a data point *(word embedding, class)* for training/testing, where *class* is of one of the following, depending on the task:

pos The Part-of-Speech of the word
headpos The PoS of the word's head
label The label of the word's dependency edge
gender* The gender of the word
case* The case of the word
number* The number of the word
tense* The tense of the word

Tasks marked with an asterisk are only carried out on words with a corresponding feature. Some of these features are absent in some languages, e.g. Basque is mostly genderless and the corpus of English we used is not annotated with morphological information. These combinations of language and feature have been omitted.

We use a one-versus-all linear classifier for two reasons: First, the feature dimensionality is relatively high. Second, and more importantly, training a linear classifier yields insights into the structure of the vector space because the classifier also serves as a tool to obtain a supervised clustering of the vector space.

Let $\mathbf{C}$ be set set of classes. A one-versus-all linear classifier learns a linear function $f_c \in \mathbb{R}^n \rightarrow \mathbb{R}$ for each class $c \in \mathbf{C}$. The classifier assigns to a vector X the best matching class based on these functions:

$$\text{class}(X) = \arg\max_{c \in \mathbf{C}} f_c(X)$$

Due to the linearity of the functions f_c, the vector space is partitioned into convex polytopes, which each represent exactly one class (see Appendix A). Therefore, the classification accuracies can also be interpreted as supervised clustering accuracies. This means that if the classifier yields a high accuracy, the members of each class are clustered in a single convex region of the vector space. We think that this is a fairly strong statement about the structure of the vector space.

To better gauge how well the embeddings are actually clustered, we use a majority baseline which classifies all elements as the one class that occurred most often during training. This is the accuracy a classifier would yield without any information and therefore the information gain obtainable by using word embeddings as features is the difference between the achieved accuracy and the baseline accuracy.

In addition to the lower bound described above, we also provide an approximate upper bound for the accuracy. Because no context information is used during classification, the word vector corresponding to a word will always be classified the same, even though the correct classification might depend on the context, e. g. the word *put* can belong to different tense classes depending on the context. Therefore, an upper bound for the classification task is to assign each word the most probable class for that word (computed on the training set). Assuming that no sparsity issues exist, embedding-based classification can yield at most accuracies as high as this approach. Note that because in reality data sparsity unfortunately does exist, this is only an approximation of the upper bound. We call this approximation *up-approx* and compute it omitting words not seen during training.

5 Experimental Setup

Evaluation was carried out on Basque, English, French, German, Hungarian, Polish, and Swedish datasets. For English, automatically labeled data was obtained by tagging and parsing a subset of the English Wikipedia dump provided by Al-Rfou et al. (2013) using TurboTagger and TurboParser (Martins et al., 2013). The Penn Treebank (Marcus et al., 1994), converted using the LTH converter (Johansson and Nugues, 2007), was used as the corresponding manually annotated resource.

For all other languages, datasets including both automatically and manually annotated data provided as part of the Shared Task on parsing morphologically rich languages (Seddah et al., 2014) were used.[2]

For all languages, we trained embeddings on the automatically labeled data using the approaches described in Section 3, with different window sizes (5 and 11, where applicable) and dimensions (10, 100, 200). The rare word limit was set to five words occurrences. **brown** was only trained with 1024 clusters equaling about 10 dimensions, as the number of clusters can not be increased to generate higher-dimensional embeddings. **dep** was not evaluated on French because the French automatically labeled dataset lacks dependency information.

6 Results

Figure 1 a) shows the accuracies for the evaluated word embeddings on all tasks for the different languages. The results were obtained using the best-performing hyperparameters (200 dimensions for all, window size = 5 for **cca**, **cbow** and **skip**, window size = 11 for **GloVe**, compare Table 1).

All embeddings capture the PoS well. To a lesser degree, the dependency label and head PoS can also be recovered. The better-performing embeddings achieve results near the approximate upper bound for all tasks.

The embeddings also mostly cluster well with respect to tense, number, gender, and case, with tense showing the best correlations. For some of these tasks, the baseline is however fairly high because the number of classes is lower.

[2]Basque: (Aduriz et al., 2003; Aldezabal et al., 2008), French: (Abeillé et al., 2003; Candito et al., 2010), German: (Brants et al., 2002; Seeker and Kuhn, 2012), Hungarian: (Csendes et al., 2005; Vincze et al., 2010), Polish: (Woliński et al., 2011; Świdziński and Woliński, 2010; Wróblewska, 2012), Swedish: (Nivre et al., 2006)

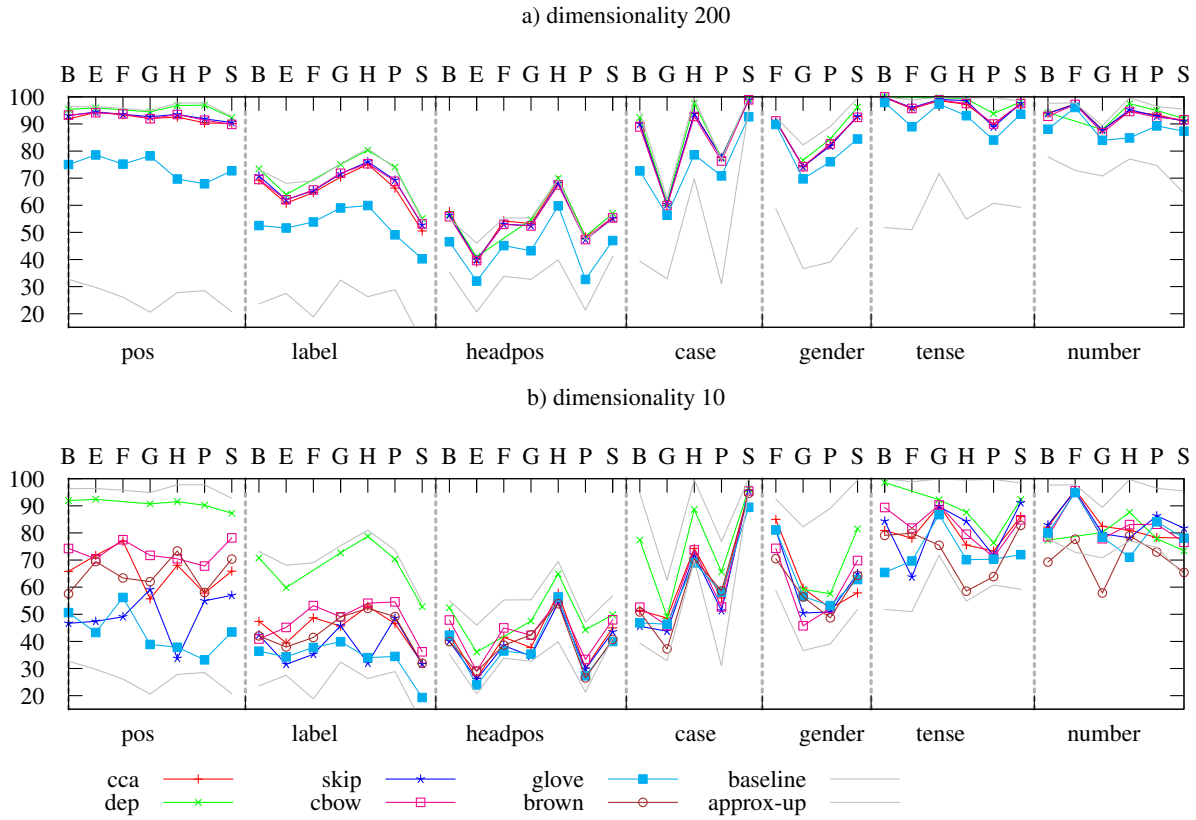

Figure 1: Results with window = 5 (for cbow, cca & skip) / 11 (for GloVe) for **B**asque, **E**nglish, **F**rench, **G**erman, **H**ungarian, **P**olish, **S**wedish. Note: brown is only present in b).

w	d	cca	skip	cbow	dep	GloVe
5	200	80.41	80.69	80.42	82.35	70.05
	100	−1.38	−1.16	−3.31	−0.39	−2.24
	10	−18.06	−22.92	−16.18	−8.38	−16.12
11	200	−1.31	−0.04	−0.05	n/a	+0.57
	100	−3.56	−1.16	−1.17	n/a	−1.73
	10	−23.51	−22.94	−16.34	n/a	−15.64

Table 1: Mean accuracy across tasks for **d**imension=200 and **w**indow=5, and change in mean accuracy when deviating, measured in percentage points. **dep** has no window parameter.

cbow, **cca** and **skip** perform nearly identical, while **dep** performs slightly better. Interestingly, **GloVe** performs consistently worse than all other embeddings, contrary to the findings published in Pennington et al. (2014), but in line with Stratos et al. (2015). **dep** performs best on nearly all tasks, which may indicate that dependency-based context is not only beneficial for preserving dependency-related information, but also for morphology.

This finding is even more pronounced in the evaluation using only ten dimensions (Figure 1 b)): While **dep** can capture the different aspects tested for nearly as well as with 200 dimensions, the other embeddings suffer larger degradations, especially for PoS and label prediction. **cbow** seems to be able to cope better with low dimensionality than **skip**, although they perform nearly identical on the high dimensionality tasks. **brown** behaves similar to the other approaches despite being quite different algorithmically and only producing low-granular data (with values for each dimension being either 1, 0, or −1). Note that results near the baseline signify that the embeddings yield only minimal benefit since the baseline does not use any features at all.

Table 1 gives an overview of the average change in accuracy when changing hyperparameters. Using 200 dimensions instead of 100 is beneficial for all word embeddings. The difference is however not nearly as pronounced as between ten and 100 dimensions. **skip** and **cbow** yield slightly better results with a window of five, whereas for **GloVe** a larger window is advantageous. **dep** achieves both the highest average score and has the lowest degradation when lowering the dimensionality.

Bansal et al. (2014) evaluate word embeddings

wrt. how they cluster along PoS tags. They first divide the embeddings into 1000 clusters using k-means and then associate each cluster with a PoS tag. They report a clustering accuracy of 81.1% for $w = 11$ and 85.8% for $w = 5$ using **skip**. Our results, however, show an accuracy of 94.4% and 94.4%, respectively, i.e. no such difference. That means that the PoS are still mostly linearly separable with larger window sizes. The differences observed by them could result from information getting lost during clustering.

7 Conclusions

Word embeddings are able to capture a range of syntactic and morphological information. They align especially well with the word's part of speech. With a high dimensionality, most embeddings perform similarly, with **GloVe** performing on average ten percentage points worse. With a low dimensionality, the differences become more pronounced and **dep** is the clear choice for applications where using high-dimensional vectors is not feasible and a correlation to the features tested in this paper is wanted.

We have shown that the different word embedding algorithms behave similar over a variety of languages and perform well relative to the task's upper bounds.

The evaluation approach proposed yields insights into the usefulness of embeddings for syntax-related tasks, works on a wide variety of languages and avoids inaccuracies introduced when employing unsupervised clustering for evaluation. We hope that this evaluation approach will be useful for evaluating future embedding techniques.

The software to replicate the experiments for this paper is available on `http://arne.chark.eu/emnlp2015`.

A Proof: Convexity of regions

To show that a one-versus-all classifier generates exactly one convex polytope for each class, we have to show that for any two points belonging to a class, each point between them belongs to the same class.

Let $c \in \mathbf{C}$ be a class and $\mathbf{r}_c \subseteq \mathbb{R}^n$ be the region(s) of c in the vector space[3] , i. e. where the following holds true:

$$f_c > f_o \; \forall o \in \mathbf{C} \setminus c$$

[3]the vector space is assumed to have one dimension for the bias.

Let $x, y \in \mathbf{r}_c$ be two points classified into c. Then the following statement needs to be true:

$$z \in \mathbf{r}_c, \;\; z := (1 - \lambda)x + \lambda y \quad \forall \lambda \in [0, 1]$$

Assume that $z \notin \mathbf{r}_c$. Then, by definition, $f_o(z) > f_c(z)$ for some $o \in \mathbf{C} \setminus c$. We can substitute z with its definition:

$$f_o((1 - \lambda)x + \lambda y) > f_c((1 - \lambda)x + \lambda y)$$

And therefore due to the linearity of f_o and f_c:

$$(1 - \lambda)f_o(x) + \lambda f_o(y) > (1 - \lambda)f_c(x) + \lambda f_c(y)$$

But this cannot be, as by definition, $f_o(x) < f_c(x)$ and $f_o(y) < f_c(y)$. Therefore, there is only one region for c and that region is convex. □

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. Building a treebank for french. In Anne Abeillé, editor, *Treebanks*. Kluwer, Dordrecht.

I. Aduriz, M. J. Aranzabe, J. M. Arriola, A. Atutxa, A. Díaz de Ilarraza, A. Garmendia, and M. Oronoz. 2003. Construction of a Basque dependency treebank. In *TLT-03*, pages 201–204.

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

I. Aldezabal, M.J. Aranzabe, A. Diaz de Ilarraza, and K. Fernández. 2008. From dependencies to constituents in the reference corpus for the processing of Basque. In *Procesamiento del Lenguaje Natural, nž 41 (2008)*, pages 147–154. XXIV edición del Congreso Anual de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN).

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 809–815, Baltimore, Maryland, June. Association for Computational Linguistics.

Jan Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In Eric P. Xing and Tony Jebara, editors, *Proceedings of The 31st International Conference on Machine Learning*, JMLR Workshop and Conference Proceedings, pages 1899–1907.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In Erhard Hinrichs and Kiril Simov, editors, *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT 2002)*, pages 24–41, Sozopol, Bulgaria.

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

Marie Candito, Benoit Crabbé, and Pascal Denis. 2010. Statistical French dependency parsing: Treebank conversion and first results. In *Proceedings of LREC*, Valletta, Malta.

Dóra Csendes, János Csirik, Tibor Gyimóthy, and András Kocsor. 2005. The Szeged treebank. In *Proceedings of the 8th International Conference on Text, Speech and Dialogue (TSD)*, Lecture Notes in Computer Science, pages 123–132, Berlin / Heidelberg. Springer.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131, January.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In Kadri Muischnek Joakim Nivre, Heiki-Jaan Kaalep and Mare Koit, editors, *Proceedings of the 16th Nordic Conference of Computational Linguistics NODALIDA-2007*, pages 105–112, University of Tartu, Tartu.

Arne Köhn, U Chun Lao, AmirAli B Zadeh, and Kenji Sagae. 2014. Parsing morphologically rich languages with (mostly) off-the-shelf software and word vectors. In *Proceedings of the 2014 Shared Task of the COLING Workshop on Statistical Parsing of Morphologically Rich Languages*.

Tao Lei, Yu Xin, Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2014. Low-rank tensors for scoring dependency structures. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1381–1391, Baltimore, Maryland, June. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado, May–June. Association for Computational Linguistics.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114–119, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andre Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 617–622, Sofia, Bulgaria, August.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of LREC*, pages 1392–1395, Genoa, Italy.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 78–86, Ann Arbor, Michigan, June. Association for Computational Linguistics.

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

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho Choi, Matthieu Constant, Richárd Farkas, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco

Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiorkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clérgerie. 2014. Overview of the SPMRL 2014 shared task on parsing morphologically rich languages. In *Notes of the SPMRL 2014 Shared Task on Parsing Morphologically-Rich Languages*, Dublin, Ireland.

Wolfgang Seeker and Jonas Kuhn. 2012. Making ellipses explicit in dependency conversion for a german treebank. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3132–3139, Istanbul, Turkey. European Language Resources Association (ELRA).

Karl Stratos, Michael Collins, and Daniel Hsu. 2015. Model-based word embeddings from decompositions of count matrices. In *Proceedings of ACL*.

Marek Świdziński and Marcin Woliński. 2010. Towards a bank of constituent parse trees for Polish. In *Proceedings of Text, Speech and Dialogue*, pages 197–204, Brno, Czech Republic.

Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010. Hungarian dependency treebank. In *Proceedings of LREC*, Valletta, Malta.

Marcin Woliński, Katarzyna Głowińska, and Marek Świdziński. 2011. A preliminary version of Składnica—a treebank of Polish. In *Proceedings of the 5th Language & Technology Conference*, pages 299–303, Poznań, Poland.

Alina Wróblewska. 2012. Polish Dependency Bank. *Linguistic Issues in Language Technology*, 7(1):1–15.