Using pseudo-senses for improving the extraction of synonyms from word embeddings

Olivier Ferret
CEA, LIST, Vision and Content Engineering Laboratory,
Gif-sur-Yvette, F-91191 France.
olivier.ferret@cea.fr

Abstract

The methods proposed recently for specializing word embeddings according to a particular perspective generally rely on external knowledge. In this article, we propose *Pseudofit*, a new method for specializing word embeddings according to semantic similarity without any external knowledge. *Pseudofit* exploits the notion of pseudo-sense for building several representations for each word and uses these representations for making the initial embeddings more generic. We illustrate the interest of *Pseudofit* for acquiring synonyms and study several variants of *Pseudofit* according to this perspective.

1 Introduction

The interest aroused by word embeddings in Natural Language Processing, especially for neural models, has led to propose methods for creating them from texts (Mikolov et al., 2013; Pennington et al., 2014) but also for specializing them according to a particular viewpoint. This viewpoint generally comes in the form of set of lexical relations. For instance, Kiela et al. (2015) specialize word embeddings towards semantic similarity or relatedness by relying either on synonyms or free lexical associations. Methods such as Retrofitting (Faruqui et al., 2015), Counter-fitting (Mrkšić et al., 2016) or PARAGRAM (Wieting et al., 2015) fall within the same framework.

The specialization of word embeddings can also come from the way they are built. For instance, Levy and Goldberg (2014) bring word embeddings towards similarity rather than relatedness by using dependency-based distributional contexts rather than linear bag-of-word contexts. Finally, some methods aim at improving word embeddings but without a clearly defined orientation, such as the All-but-the-Top method (Mu, 2018), which focuses on dimensionality reduction, or (Vulić et al., 2017), which exploits morphological relations.

In this article, we propose *Pseudofit*, a method that improves word embeddings without external knowledge and focuses on semantic similarity and synonym extraction. The principle of *Pseudofit* is to exploit the notion of pseudo-sense coming from word sense disambiguation for building representations accounting for distributional variability and to create better word embeddings by bringing these representations closer together. We show the interest of *Pseudofit* and its variants through both intrinsic and extrinsic evaluations.

2 Method

The distributional representation of a word varies from one corpus to another. Without even taking into account the plurality of meanings of a word, this variability also exists inside any corpus C, even if it is quite homogeneous: the distributional representations of a word built from each half of C, C_1 and C_2, are not identical. However, from the more general viewpoint of its meaning, they should be identical, or at least very close, and their differences be considered as incidental. Following this perspective, a representation resulting from the convergence of the representations built from C_1 and C_2 should be more generic and show better semantic similarity properties.

The method we propose, *Pseudofit*, formalizes this approach through the notion of pseudo-sense. This notion is related to the notion of pseudo-word introduced in the field of word sense disambiguation by Gale et al. (1992) and Schütze (1992). A pseudo-word is an artificial word resulting from the clustering of two or more different words, each of them being considered as one pseudo-sense of

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 351–357
Melbourne, Australia, July 15 - 20, 2018.

the pseudo-word. *Pseudofit* adopts the opposite viewpoint. For each word w, more precisely nouns in our case, it splits arbitrarily its occurrences into two sets: the occurrences of one set are labeled as pseudo-sense w_1 while the occurrences of the other set are labeled as pseudo-sense w_2. A distributional representation is built for w, w_1 and w_2 under the same conditions, with a neural model in our case. The second stage of *Pseudofit* adapts *a posteriori* the representation of w according to the convergence of the representations of w_1 and w_2. This adaptation is performed by exploiting the similarity relations between w, w_1 and w_2 in the context of a word embedding specialization method. By considering simultaneously w, w_1 and w_2, *Pseudofit* benefits from both the variations between the representations of w_1 and w_2 and the quality of the representation of w, since it is built from the whole C while the two others are built from half of it.

2.1 Building of Word Embeddings

The first stage of *Pseudofit* consists in building a distributional representation of each word w and its two pseudo-senses w_1 and w_2. The starting point of this process is the generation of a set of distributional contexts for each occurrence of w. Classically, this generation is based on a linear fixed-size window centered on the considered occurrence. The specificity of *Pseudofit* is that contexts are generated both for the target word and one of its pseudo-sense. The pseudo-sense changes from one occurrence of w to the following, leading to the same frequency for w_1 and w_2. The generation of such contexts with a window of 3 words (before and after the target word *policeman*) is illustrated here for the following sentence:

A *policeman*$_1$ was arrested by another *policeman*$_2$.

TARGET	CONTEXTS
policeman	{a, be, arrest (2), by (2), another}
policeman$_1$	{a, be, arrest, by}
policeman$_2$	{another, by, arrest}

This sentence, which is voluntarily artificial, shows how three different contexts are built for a word in a corpus: one context (first line) is built from all the occurrences of the target word; a second one (second line) is built from half of the occurrences of the target word, representing its first pseudo-sense, while the third context (last line) is built from the other half of the occurrences of the target word, representing its second pseudo-sense. The generated contexts are then used for building word embeddings. More precisely, we adopt the variant of the Skip-gram model (Mikolov et al., 2013) proposed by Levy and Goldberg (2014), which can take as input arbitrary contexts.

2.2 Convergence of Word Representations

The second stage of *Pseudofit* brings the representations of each target word w and its pseudo-senses w_1 and w_2 closer together. This convergence aims at producing a more general representation of w by erasing the differences between the representations of w, w_1 and w_2, which are assumed to be incidental since these representations refer by nature to the same object.

The implementation of this convergence process relies on the PARAGRAM algorithm, which takes as inputs word embeddings and a set of binary lexical relations accounting for semantic similarity. PARAGRAM gradually modifies the input embeddings for bringing closer together the vectors of the words that are part of similarity relations. This adaptation is controlled by a kind of regularization that tends to preserve the input embeddings. This twofold objective consists more formally in minimizing the following objective function by stochastic gradient descent:

$$\sum_{(x_1,x_2)\in\mathcal{L}_i} \max(0, \delta + \mathbf{x}_1\mathbf{t}_1 - \mathbf{x}_1\mathbf{x}_2) + \max(0, \delta + \mathbf{x}_2\mathbf{t}_2 - \mathbf{x}_1\mathbf{x}_2) + \lambda \sum_{\mathbf{x}_i \in V(\mathcal{L}_i)} \left\| \mathbf{x}_i^{init} - \mathbf{x}_i \right\|^2 \quad (1)$$

where the first sum expresses the convergence of the vectors according to the similarity relations while the second sum, modulated by the λ parameter, corresponds to the regularization term.

The specificity of PARAGRAM, compared to methods such as Retrofitting, lies in its adaptation term. While it logically tends to bring closer together the vectors of the words that are part of similarity relations (attracting term $\mathbf{x}_1\mathbf{x}_2$), it also pushes them away from the vectors of the words that are not part these relations (repelling terms $\mathbf{x}_1\mathbf{t}_1$ and $\mathbf{x}_2\mathbf{t}_2$). More precisely, the relations are split into a set of mini-batches $\mathcal{L}_i$. For each word (vector $\mathbf{x}_i$) of a relation, a word (vector $\mathbf{t}_j$) outside the relation is selected among the words of the mini-batch of the current relation in such a way that $\mathbf{t}_j$ is the closest word to $\mathbf{x}_i$ according to the *Cosine* measure, which represents the most discriminative option. δ is the margin between the attracting and repelling terms.

	INITIAL	Pseudofit	Retrofit.	Counter-fit.
SimLex-999	49.5	51.2	49.6	49.5
MEN	78.3	79.9	77.4	77.2
MTurk 771	65.6	68.0	65.0	64.9

Table 1: Intrinsic evaluation of *Pseudofit* ($\times 100$)

The application of PARAGRAM to the embeddings resulting from the first stage of *Pseudofit* exploits the fact that a word and its pseudo-words are supposed to be similar. Hence, for each word w, three similarity relations are defined and used by PARAGRAM for adapting the initial embeddings: (w, w_1), (w, w_2) et (w_1, w_2). Finally, only the representations of words w are exploited since they are built from a corpus that is twice as large as the corpus used for pseudo-words.

3 Experiments

3.1 Experimental Setup

For implementing *Pseudofit*, we randomly select at the level of sentences a 1 billion word subpart of the Annotated English Gigaword corpus (Napoles et al., 2012). This corpus is made of news articles in English processed by the Stanford CoreNLP toolkit (Manning et al., 2014). We use this corpus under its lemmatized form. The building of the embeddings are performed with *word2vecf*, the adaptation of *word2vec* from (Levy and Goldberg, 2014), with the best parameter values from (Baroni et al., 2014): minimal count=5, vector size=300, window size=5, 10 negative examples and 10^{-5} for the subsampling probability of the most frequent words. For PARAGRAM, we adopt most of the parameter values from (Vulić et al., 2017): $\delta = 0.6$ and $\lambda = 10^{-9}$, with the AdaGrad optimizer (Duchi et al., 2011) and 50 epochs[1]. Retrofitting and Counter-fitting are used with the parameter values specified respectively in (Faruqui et al., 2015) and (Mrkšić et al., 2016).

3.2 Evaluation of *Pseudofit*

Our first evaluation of *Pseudofit* at word level is a classical intrinsic evaluation consisting in measuring for a set of word pairs the Spearman's rank correlation between human judgments and the similarity of these words computed from their embeddings by the *Cosine* measure. This evaluation is performed for the nouns of three large enough reference datasets: SimLex-999 (Hill et al., 2015), MEN (Bruni et al., 2014) and MTurk-771 (Halawi et al., 2012). Table 1 clearly shows that *Pseudofit* significantly[2] improves the initial embeddings for the three datasets. By contrast, it also shows that replacing PARAGRAM with Retrofitting or Counter-fitting, two other reference methods for specializing embeddings, does not lead to comparable improvements and can even degrade results.

[1]We used the implementation of PARAGRAM provided by https://github.com/nmrksic/attract-repel.

method	**R**$_{prec.}$	**MAP**	**P@1**	**P@2**	**P@5**
INITIAL	13.0	15.2	18.3	13.1	7.7
Pseudofit	+2.5	+3.3	+3.0	+2.5	+1.8
Retrofitting	−0.5	−0.6	−0.6	−0.2†	−0.3
Counter-fitting	−0.6	−0.8	−0.6	−0.5	−0.4

Table 2: Evaluation of *Pseudofit* for synonym extraction (differences / INITIAL, $\times 100$)

Our second evaluation, which is our main focus, is a more extrinsic task consisting in extracting synonyms[3]. This extraction is performed by ranking a set of candidate synonyms for each target word according to the similarity, computed here by the *Cosine* measure, of their embeddings. We evaluate the relevance of this ranking as in Information Retrieval with R-precision (R$_{prec.}$), MAP (Mean Average Precision) and precisions at various ranks (P@r). Our reference is made up of the synonyms of WordNet (Miller, 1990) while both our target words and candidate synonyms are made up of the nouns with more than ten occurrences in each half of our corpus, which represents 20,813 nouns.

Table 2 gives the result of this second evaluation for 11,481 nouns with synonyms in WordNet among our 20,813 targets. As in the first evaluation, *Pseudofit* significantly[4] outperforms the initial embeddings. Moreover, replacing PARAGRAM with Retrofitting or Counter-fitting leads to a systematic decrease of results, which emphasizes the importance of the repelling term of PARAGRAM. This term probably prevents the representation of a word from being changed too much by its pseudo-senses, which are interesting variants in terms of representations but were built from half of the corpus only.

[2]The statistical significance of differences are judged according to a two-tailed Steiger's test with p-value < 0.01 with the R package *cocor* (Diedenhofen and Musch, 2015).

[3]The TOEFL test, which is close to our task, is considered sometimes as extrinsic and sometimes as intrinsic.

[4]The significance of differences are judged according to a paired Wilcoxon test with the following notation: nothing if $p <= 0.01$, † if $0.01 < p \leq 0.05$ and ‡ if $p > 0.05$.

method	$R_{prec.}$	MAP	P@1	P@2	P@5
INITIAL$_{high}$	15.4	17.7	22.6	16.4	9.7
INITIAL$_{low}$	9.4	11.5	11.8	8.1	4.7
Pseudofit$_{high}$	+0.7	+1.1	+0.3‡	+0.8	0.9
Pseudofit$_{low}$	+5.3	+6.7	+7.0	+5.2	+3.1

Table 3: Evaluation of *Pseudofit* for synonym extraction according to the frequency (*high* or *low*) of the target words (differences / INITIAL, $\times 100$)

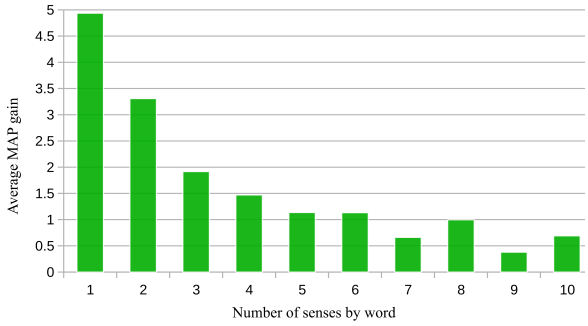

Figure 1: Gain brought by *Pseudofit* for MAP according to the ambiguity of the target word

Finally, we performed a finer analysis of these results according to the frequency and the degree of ambiguity of the target words. Concerning frequency, Table 3 shows that *Pseudofit* is particularly efficient for the lower half of the target words in terms of frequency, with a large increase of 5.3 points for R-precision, 6.7 points for MAP, 7.0 points for P@1 and 5.2 points for P@2 while the largest increase for the higher half of the target words is equal to 1.1 points for MAP.

One possible explanation of this gap between high and low frequency words is linked to the degree of ambiguity of words: high frequency words are more likely to be polysemous and *Pseudofit* does not take into account the polysemy of words. Figure 1 tends to confirm this hypothesis by showing that the improvement brought by *Pseudofit* for a word is inversely proportional to its ambiguity as estimated by its number of senses in WordNet[5].

3.3 Variants of *Pseudofit*

We defined and tested several variants of *Pseudofit*. The first one, *Pseudofit max*, focuses on the strategy for selecting $\{\mathbf{t}_j\}$ in PARAGRAM. The results of Table 1, as those of (Mrkšić et al., 2017), are obtained with a setting where half of $\{\mathbf{t}_j\}$ are selected randomly. In *Pseudofit max*, all $\{\mathbf{t}_j\}$ are selected according to their similarity with $\{\mathbf{x}_i\}$.

[5]Words with at most 10 senses cover 98.9% of the nouns of our evaluation.

Variant	$R_{prec.}$	MAP	P@1	P@2	P@5
Pseudofit	15.5	18.5	21.3	15.6	9.5
max	+0.2‡	+0.3	+0.3†	+0.2†	+0.1
3 pseudo-senses	+0.2‡	+0.2	+0.4†	+0.2‡	+0.0‡
context	+0.4†	+0.3‡	+0.5†	+0.2‡	+0.0‡
fus-average	+0.2†	+0.3	+0.4	+0.2†	+0.1
fus-add	+0.0‡	+0.0	+0.2‡	+0.1‡	+0.1†
fus-max-pool	+0.2‡	+0.3	+0.4	+0.2	+0.2
max+fus-max-pool	+0.4	+0.5	+0.5	+0.4	+0.2

Table 4: Evaluation of *Pseudofit*'s variants (differences / *Pseudofit*, $\times 100$)

The second variant, *Pseudofit 3 pseudo-senses*, aims at determining if increasing the number of pseudo-senses, from two to three at first, can have a positive impact on results.

The third variant, *Pseudofit context*, tests the interest of defining pseudo-senses for the words of distributional contexts. In this configuration, pseudo-senses are defined for all nouns, verbs and adjectives with more than 21 occurrences in the corpus, which corresponds to a minimal frequency of 10 in each half of the corpus.

Finally, similarly to the second variant, the last variant, *Pseudofit fus-**, adds a supplementary representation of the target word. However, this representation is not an additional pseudo-sense but an aggregation of its already existing pseudo-senses, which can be viewed as another global representation of the target word. Three aggregation methods are considered: *Pseudofit fus-addition* performs an elementwise addition of the embeddings of pseudo-senses, *Pseudofit fus-average* computes their mean while *Pseudofit fus-max-pooling* takes their maximal value.

Each presented variant outperforms the base version of *Pseudofit* but Table 4 also shows that not all variants are of equal interest. From the viewpoint of both the absolute level of their results and the significance of their difference with *Pseudofit*, *Pseudofit max* and *Pseudofit fus-max-pooling* are clearly the most interesting variants. Their combination, *Pseudofit max+fus-max-pooling*, leads to our best results and significantly outperforms *Pseudofit* for all measures. Among the *Pseudofit fus-** variants, *Pseudofit fus-max-pooling* and *Pseudofit fus-average* are close to each other and clearly exceeds *Pseudofit fus-addition*. The results of *Pseudofit 3 pseudo-senses* show that using more than two pseudo-senses by

word faces the problem of having too few occurrences for each pseudo-sense. The same frequency effect, at the level of contexts, probably explains the very limited impact of the introduction of pseudo-senses in contexts in the case of *Pseudofit context*.

3.4 Sentence Similarity

Our final evaluation, which is fully extrinsic, examines the impact of *Pseudofit* on the identification of semantic similarity between sentences. More precisely, we adopt the STS Benchmark dataset on semantic textual similarity (Cer et al., 2017). The overall principle of this task is similar to the word similarity task of our first evaluation but at the level of sentences: the similarity of a set of sentence pairs is computed by the system to evaluate and compared with a correlation measure, the Pearson correlation coefficient, against a gold standard produced by human annotators.

This framework is interesting for the evaluation of *Pseudofit* because the computation of the similarity of a pair of sentences can be achieved by unsupervised approaches based on word embeddings in a very competitive way, as demonstrated by (Hill et al., 2016). More precisely, the approach we adopt is a classical baseline that composes the embeddings of the plain words of each sentence to compare by elementwise addition and computes the *Cosine* measure between the two resulting vectors. For building the representation of a sentence, we compare the use of our initial embeddings with that of the embeddings produced by *Pseudofit max+fus-max-pooling*, the best variant of *Pseudofit*. For this experiment, pseudo-senses are distinguished not only for nouns but more generally for all nouns, verbs and adjectives with more than 21 occurrences in the corpus.

Table 5 shows the result of this evaluation for the 1,379 sentence pairs of the test part of the STS Benchmark dataset. As for the two previous evaluations, the use of the embeddings modified by *Pseudofit* leads to a significant improvement of results[6] compared to the initial embeddings, which demonstrates that the improvement at word level can be transposed at a larger scale. Table 5 also shows four reference results from (Cer et al., 2017): the lowest and the best baselines based on averaged word embeddings (Skip-gram and GloVe respectively), which are very close to our approach, and the best (Conneau et al., 2017) and the lowest (Duma and Menzel, 2017) unsupervised systems. Although our goal is not to compete with the best systems, it is interesting to note that our results are in line with the state of the art since they significantly outperform the two baselines and the lowest unsupervised system as well as other unsupervised systems mentioned in (Cer et al., 2017).

method	$\rho \times 100$
INITIAL	63.2
Pseudofit max+fus-max-pooling	66.0
(Cer et al., 2017)	
Best baseline (averaged embeddings)	56.5
Lowest baseline (averaged embeddings)	40.6
Best unsupervised system	75.8
Lowest unsupervised system	59.2

Table 5: Evaluation of *Pseudofit* for identifying sentence similarity

4 Conclusion and Perspectives

In this article, we presented *Pseudofit*, a method that specializes word embeddings towards semantic similarity without external knowledge by exploiting the variability of distributional contexts. This method can be described as hybrid since it operates both before and after the building of word embeddings. A set of intrinsic and extrinsic evaluations demonstrates the interest of the word embeddings produced by *Pseudofit* and its variants, with a particular emphasis on the extraction of synonyms.

In the presented work, the principles underlying *Pseudofit*, in particular the generation and convergence of different representations of a word, were tested only within the same corpus. In conjunction with the work about word meta-embeddings (Yin and Schütze, 2016), it would be interesting to apply these principles to representations built from several corpora, like (Mrkšić et al., 2017) for different languages.

Acknowledgments

This work has been partially funded by French National Research Agency (ANR) under project ADDICTE (ANR-17-CE23-0001). The author thanks the anonymous reviewers for their valuable comments.

[6] With the same evaluation of statistical significance as for word similarity.

References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 238–247, Baltimore, Maryland.

Elia Bruni, N Tram, Marco Baroni, et al. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 670–680, Copenhagen, Denmark.

Birk Diedenhofen and Jochen Musch. 2015. cocor: A Comprehensive Solution for the Statistical Comparison of Correlations. *PLOS ONE*, 10(4):1–12.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Mirela-Stefania Duma and Wolfgang Menzel. 2017. SEF@UHH at SemEval-2017 Task 1: Unsupervised Knowledge-Free Semantic Textual Similarity via Paragraph Vector. In *11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 170–174, Vancouver, Canada.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2015)*, pages 1606–1615, Denver, Colorado.

William A Gale, Kenneth W Church, and David Yarowsky. 1992. Work on statistical methods for word sense disambiguation. In *AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 54–60.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale Learning of Word Relatedness with Constraints. In *18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*, pages 1406–1414. ACM.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning Distributed Representations of Sentences from Unlabelled Data. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)*, pages 1367–1377, San Diego, California.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing Word Embeddings for Similarity or Relatedness. In *2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 2044–2048, Lisbon, Portugal.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 302–308, Baltimore, Maryland.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), system demonstrations*, pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR 2013, workshop track*.

George A. Miller. 1990. WordNet: An On-Line Lexical Database. *International Journal of Lexicography*, 3(4).

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting Word Vectors to Linguistic Constraints. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)*, pages 142–148, San Diego, California.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Gašić Milica, Anna Korhonen, and Steve Young. 2017. Semantic Specialization of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324.

Jiaqi Mu. 2018. All-but-the-Top: Simple and Effective Postprocessing for Word Representations. In *Sixth International Conference on Learning Representations (ICLR 2018), poster session*, Vancouver, Canada.

Courtney Napoles, Matthew R. Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *NAACL Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543, Doha, Qatar.

Hinrich Schütze. 1992. Dimensions of meaning. In *1992 ACM/IEEE conference on Supercomputing*, pages 787–796. IEEE Computer Society Press.

Ivan Vulić, Nikola Mrkšić, Roi Reichart, Diarmuid Ó Séaghdha, Steve Young, and Anna Korhonen. 2017. Morph-fitting: Fine-Tuning Word Vector Spaces with Simple Language-Specific Rules. In *55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 56–68, Vancouver, Canada.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From Paraphrase Database to Compositional Paraphrase Model and Back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

Wenpeng Yin and Hinrich Schütze. 2016. Learning Word Meta-Embeddings. In *54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1351–1360, Berlin, Germany.