Word Embeddings as Features for Supervised Coreference Resolution

Iliana Simova
Saarland University
Saarbrücken, Germany
ilianas@coli.uni-saarland.de

Hans Uszkoreit
Language Technology Lab, DFKI
Alt-Moabit 91c, Berlin, Germany
uszkoreit@dfki.de

Abstract

A common reason for errors in coreference resolution is the lack of semantic information to help determine the compatibility between mentions referring to the same entity. Distributed representations, which have been shown successful in encoding relatedness between words, could potentially be a good source of such knowledge. Moreover, being obtained in an unsupervised manner, they could help address data sparsity issues in labeled training data at a small cost. In this work we investigate whether and to what extend features derived from word embeddings can be successfully used for supervised coreference resolution. We experiment with several word embedding models, and several different types of embedding-based features, including embedding cluster and cosine similarity-based features. Our evaluations show improvements in the performance of a supervised state-of-the-art coreference system.

1 Introduction

Coreference resolution is the task of automatically identifying expressions in a text which refer to the same real-world entity. It is an important intermediate step for a number of Natural Language Processing (NLP) applications which depend on text understanding, such as machine translation, relation extraction, and question answering, to name a few.

There is a wide range of approaches for solving coreference resolution. Some successful systems rely on rules and heuristics (Lee et al., 2011), others employ a variety of machine learning models (Martschat et al., 2015; Björkelund and Kuhn, 2014). One aspect they have in common is the recognition that the process could benefit from some form of lexical and encyclopedic knowledge. For instance, lexical relations such as synonymy and hyperonymy (e.g., *a capital* is *a city*), world knowledge (e.g., *France* is *a country*), and even attributive knowledge (e.g., *countries* have *areas* and *population*) could all prove helpful for the task.

Durrett and Klein (2013) have shown that to achieve state-of-the-art performance in coreference resolution, one does not need complicated heuristic-driven features. However, they identify the task of including semantics into the process as an "uphill battle". Many works explore sources of semantic knowledge for the task. Some include mining unannotated data (Bean and Riloff, 2004; Mohit Bansal, 2012), structured lexical databases (Daumé III and Marcu, 2005), and knowledge bases (Rahman and Ng, 2011). In this work we explore a different source of semantic knowledge, namely distributed word representations.

Word embeddings are representations of word meaning taking the perspective that the sense of a word is defined by the company it keeps, or its context words. In some models, these representations are learned as a prediction task using neural networks (Mikolov et al., 2013), other approaches use dimensionality reduction on a co-occurrence matrix (Pennington et al., 2014). The resulting representations have been shown to encode word relatedness and similarity. Depending on the context selected to represent a word's sense, this similarity can take different forms (Levy and Goldberg, 2014). Such representations are typically learned from large collections of text in an unsupervised manner. If we employ them to a supervised task, they could also prove useful in addressing some data sparsity issues in the training data. Therefore they are an attractive source of semantic

Proceedings of Recent Advances in Natural Language Processing, pages 686–693,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_088

knowledge, which can be obtained at a small cost and be beneficial in multiple ways.

In this work we explore the usefulness of features derived from word embeddings for the task of coreference resolution. We experiment with several general purpose word embedding models, and several different types of features derived from word embeddings, including embedding cluster and cosine similarity-based features.

2 Related Work

Word embeddings generally serve as input to deep learning architectures. Several works have applied such techniques successfully to the task of coreference resolution (Wiseman et al., 2016; Clark and Manning, 2016). While such systems reach state-of-the-art performance without the need for elaborate handcrafted features, complexity is added in terms of the model. In this work we investigate whether word embeddings could also be beneficial when used in another simpler way and a standard setting - in the form of features for supervised learning. To the best of our knowledge this is the first work to apply such features to supervised coreference resolution.

Several works have explored the application of word embedding features to supervised learning systems for other NLP tasks.

Turian et al. (2010) compare several unsupervised word representation models for Named Entity Recognition (NER) and chunking. The authors find that each of them improves on state-of-the-art supervised baselines. Yu et al. (2013) further explore other ways of encoding word embedding information at a lower computational cost by performing prior clustering of the embeddings and inducing a cluster label feature.

In (Guo et al., 2014), word embedding-based features are studied in relation to the task of NER. The work further explores several different ways of deriving features from word embeddings - 1) by creating a binarized version of the embedding to consider features with strong opinions on each dimension, 2) by clustering of the embedding via k-means and including the cluster label as a feature, and 3) as a distributional prototype feature. In the latter, a few examples are automatically selected to represent each type of named entity, and then cosine similarity is calculated between the embeddings of the word in question and these prototypes. All word embedding-based features brought improvements to the baseline, and over the direct use of the embedding. Moreover, a combination of the features led to a greater improvement in performance, indicating that the knowledge represented by them is not overlapping completely.

Similarly to Guo et al. (2014), in this work we experiment with several different word embedding-based features. We employ prior clustering of the word embeddings and include the obtained cluster label as a feature, with the expectation that it would encourage semantic compatibility between candidate coreferring expressions. One difference here lies in choice of clustering algorithm - spherical k-means was selected due to its use of cosine similarity as a distance metric as opposed to Euclidean distance which is employed in standard k-means. We consider this metric more suitable when comparing word embeddings, and expect that its use would result in more meaningful clusters. In addition to also directly using the original word embedding as a feature, we experiment with reduced version of it. While a prototype-based feature could also prove useful for this task, we plan to investigate it in future work.

3 Experimental Setup

This section offers more details on the way in which we use word embeddings as features for supervised learning, as well as on the word embedding models selected for the current experiments. We further present the baseline coreference system.

3.1 Word Embedding Models

There is a fairly big selection of available pre-trained general purpose word embedding models. As a first step towards designing word embedding-based features, we have selected three of them with varying properties. These include: the word2vec embedding (Mikolov et al., 2013) trained on part of the Google News dataset, with a vocabulary of 3 million words and phrases (e.g., New_York), a GloVe embedding (Pennington et al., 2014) trained on Wikipedia and Gigaword 5 with a vocabulary of 400 thousand words, and the dependency-based word embedding (Levy and Goldberg, 2014) model trained on part of Wikipedia, with a vocabulary size of about 180 thousand words. All of the selected embeddings are of dimension 300. For simplicity we refer to them as *w2v*, *glove* and *deps*, respectively.

Our aim is not to provide a fair comparison of these models, but rather to see if some of them with their specific characteristics would prove particularly useful for our task. For instance, the first two embedding models have different types of training data. The word2vec model is trained on news data, which coincides with the majority of the data in the OntoNotes corpus, and has the largest vocabulary of the three. The GloVe model, in addition to news data present in Gigaword, also incorporates the encyclopedic knowledge of Wikipedia. The dependency embedding, on the other hand, takes a different perspective on which words in the context of a word are important for determining it's sense, and make use of dependency analysis in order to select meaningful contexts. The resulting embeddings have been shown to exhibit more functional similarity (Levy and Goldberg, 2014).

3.2 Features Derived from Word Embeddings

3.2.1 Embedding Cluster

We perform a clustering of word embeddings and assign a cluster label to the head words of each mention pair under consideration. The motivation behind this is that this clustering would provide us with a form of a semantic tag which could either ensure semantic compatibility between the mentions directly when two mentions fall into the same cluster, or be used by the supervised system to learn compatible anaphor-antecedent combinations from the training data. During clustering, each word embedding vector is treated as a single instance, and the resulting clusters consist of the words with the most similar embeddings according to a distance metric.

The selection of clustering algorithm is guided by the choice of suitable distance metric. As cosine similarity is popularly used to compare embeddings of words, our clustering algorithm of choice is spherical k-means[1].

We experiment with several different values for number of clusters in order to determine what granularity is most suitable for the task. The values include 50, 100, 500 for less fine-grained clusters, and 1k to 10k in steps of 2.5k for more fine-grained ones.

[1] https://github.com/clara-labs/spherecluster

3.2.2 Dense Embedding Features

Another way to deliver word embedding information to the coreference system is by including the embedding vector directly. In this setting each dimension of the vector of a mention's head word is a separate numeric feature. This is justified by the fact that different dimensions of each embedding can be considered as latent features encoding different properties of a word (Turian et al., 2010).

One consideration here is that by including the whole vector we increase the amount of features substantially. Therefore we also experimented with reduced versions of the original vectors by Principal Component Analysis (PCA)[2]. PCA examines the correlation between different dimensions in the word embedding vector, and identifies a smaller number of variables that best explain the original vector. The vectors were reduced to retain different amounts of variability present in the original data, which addresses over-fitting, and effectively reduces the amount of features to be added. We experiment with variance levels of 10% to 100% in steps of 10%.

3.2.3 Cosine Similarity Features

The third set of word embedding features is based on cosine similarity. Cosine similarity computes the cosine of the angle between two vectors, and in the case of word embeddings, provides a measure of relatedness between words. We estimate the similarity between the head words of each anaphor and antecedent (ANA, ANTE) in a candidate pair, as well as between their governing words (GOV_{ana}, GOV_{ante}) in the dependency tree, and the combinations ANA-GOV_{ante} and ANTE-GOV_{ana}.

This feature could prove useful in cases of pronoun resolution, where the context of the pronoun referring expression is a deciding factor. Consider the example from (Jespersen, 1949), "If the *baby* does not thrive on *raw milk*, boil *it*.", where the pronoun *it* has two candidate antecedents. The cosine similarity between the governing word of the anaphor, *boil*, and the second antecedent candidate, *milk*, is very high, as opposed to between the alternative pairing which leads to an undesirable interpretation. As GOV_{ana} and GOV_{ante} are often verbs, this feature could also prove useful when coreferring mentions carry out related actions ("During an interview he *said* [..] He further *reported* that [..]").

[2] package sklearn.decomposition.PCA

Feature set	F1
baseline	59.19
+EC$_{w2v,2.5k}$	59.56
+CS$_{glove,pca\text{-}20}$	59.49
+WE$_{glove,pca\text{-}50}$	59.67
+EC$_{glove,2.5k}$ +CS$_{glove,pca\text{-}20}$	59.68
+CS$_{deps,pca\text{-}60}$ +WE$_{glove,pca\text{-}50}$	59.66
+EC$_{glove,2.5k}$ +WE$_{glove,pca\text{-}50}$	59.66
+EC$_{glove,5k}$ +CS$_{deps,pca\text{-}90}$ +WE$_{glove}$	**59.72**

Table 1: Summary of the performance of the coreference system (CoNLL F1) with different sets of features. Individual features include: embedding cluster (EC) label, cosine similarity (CS) features, and word embedding (WE) features, as described in subsection 3.2.

When calculating the cosine similarity between two words, we use the vectors of the original word embeddings, as well as PCA-reduced versions of them.

3.3 Coreference Resolution System

The coreference resolution toolkit *cort* (Martschat et al., 2015; Martschat and Strube, 2015) was used to obtain the baseline, and extended in our further experiments. The system offers several implementations of popular coreference resolution approaches, as well as ways of visualizing and comparing the outputs of different models. In this work we employ a *mention-pair* model with *best-first* clustering. This model breaks down the task of coreference resolution into pairwise coreference decisions. The construction of coreference chains is then enabled via clustering.

The baseline employs a standard set of features, including: number, gender, various string matching and distance features, semantic class ("person", "object", "numeric"), fine type (name, definite/indefinite noun phrase, etc.), and others [3].

4 Results

4.1 Evaluation

Table 1 contains the best results achieved by the coreference resolution tool per feature and feature combination. All of the models were trained and tested on the CoNLL-2012 training and test data sets with automatic preprocessing. The evaluation metric is CoNLL-F1 score (average of MUC, B^3, and $CEAF_e$). Due to space limitations we do not provide a detailed list of results, but a discussion of the most important ones is included below.

When considering the embedding cluster feature in isolation, we observed that the best performing setting is that with the *w2v* embedding and 2.5K number of clusters. For the other two embedding models, the same amount of clusters was also most successful. We observed a drop in performance for high number of clusters (over 7.5K) with all three models. In 12% of the experiments we obtained a small drop in performance below the baseline score.

For the cosine similarity set of features, the best performance was achieved by *glove* with explained variance of 20%. Using the whole word embedding vectors to compute this feature lead to worse results for all embedding models, and the performance of the feature across different reductions was unstable, often dropping slightly below the baseline level (results were in the range 58.97-59.49, with 36% worse than the baseline).

The best result with the word embedding feature was achieved with *glove* and variance of 50%. Here we also observed that when the original embeddings were used, the performance was worse compared to the reduced versions. The models' performance ranged from 59.06 to 59.67, with 25% of them performing worse than the baseline.

The second part of Table 1 shows the best pairwise combinations of features. We experimented with combinations of some of the best performing individual features. Our goal is to determine if the knowledge they encode is complementary to each other. The most complementary feature combination is embedding cluster and cosine similarity. For the rest of the combinations we did not observe any gains in performance over the use of the best individual feature.

With a combination of three of the feature types we obtained the highest scoring model of 59.72 F1 score.

4.2 Error Analysis

As the automatic evaluation doesn't offer a very good insight on where the word embedding feature models differ with respect to the baseline model and each other, we include a more detailed recall and precision error analysis.

We use the methodology proposed in

[3] see https://github.com/smartschat/cort for a complete list

Precision Error Analysis					
Error Category	Feature Set				
ANA → ANTE	baseline	+EC$_{w2v,2.5k}$	+CS$_{glove,pca\text{-}20}$	+WE$_{glove,pca\text{-}50}$	+EC+CS+WE$_{best}$
Pron → Pron	33%	33%	34%△	35%△	33%
Pron → Name	5%	5%	5%	5%	5%
Pron → Noun	7%	7%	7%	**6%**	7%
Name → Pron	1%	2%△	2%△	1%	1%
Name → Name	21%	21%	21%	22%△	22%△
Name → Noun	1%	2%△	1%	1%	1%
Noun → Pron	1%	1%	1%	1%	1%
Noun → Name	2%	**1%**	2%	2%	2%
Noun → Noun	27%	**26%**	**25%**	**26%**	**26%**
total num. errors	4758	**4712**	**4638**	**4500**	**4565**

Table 2: Precision error analysis of some of the best models per type of anaphor and antecedent. A decrease in the amount of errors is marked in **bold**, and an increase: with the symbol △.

(Martschat and Strube, 2014). In this study, coreference chains are viewed as complete one directional graphs, following the order in which mentions occur in the text. Error analysis is then viewed as comparing graphs in terms of edges. A system's output is transformed into a maximum spanning tree using a notion from accessibility theory to select the most likely missing links needed to reach the reference graph. If an edge in the resulting graph is missing from the reference entity, this is considered a recall error. To extract precision errors, the anaphor and antecedent decisions made by a system are used in a similar manner. This representation allows for the recall and precision errors to be categorized in terms of type of anaphor and antecedent.

4.2.1 Precision Error Analysis

The results of our precision error analysis are provided in Table 2. In bold we mark all cases where a reduction of the amount of errors is visible as compared to the baseline system, while an increase in the amount of errors is marked with the symbol △. The percentages denote the proportion of errors per error type from the total number of errors a system made. For instance, in the baseline system, resolving a link between a pronoun anaphor and a pronoun antecedent ("Pron → Pron") was responsible for 33% of the total number of errors, 4758.

All of the systems manage to achieve a reduction in the total number of precision errors compared to the baseline system (see "total number of errors"). The word embedding model comes first with 258 fewer errors, followed by the best combination model with 193 fewer errors.

We observe interesting results for the category "Noun → Noun", where both the anaphor and antecedent are noun phrases. All of the systems manage to improve over the baseline, showing the influence of the semantic knowledge introduced by the word embedding features. Here surprisingly the most improvement is achieved by the cosine similarity model (25% or 105 fewer errors in raw counts).

Our initial hypothesis that the cosine similarity model would be especially useful for pronoun resolution was not supported by the evaluation results. For the category "Pron → Pron", it even lead to more precision errors.

The word embedding model is the one which stands out in terms of overall performance, but it leads to more errors in two of the categories: "Pron → Pron" and "Name → Name". For the first one, it is perhaps to be expected, as the word embedding of a pronoun is not particularly informative, given no additional context information. The latter result was surprising, as we anticipated that the model would be able to detect more aliases and name variations.

The combination model does not seem to inherit some of the negative properties of the other models, as it does not have worse performance than the baseline for most of the categories which were

Recall Error Analysis					
Error Category	Feature Set				
ANA → ANTE	baseline	+EC$_{w2v,2.5k}$	+CS$_{glove,pca\text{-}20}$	+WE$_{glove,pca\text{-}50}$	+EC+CS+WE$_{best}$
Pron → Pron	7%	7%	7%	7%	7%
Pron → Name	10%	10%	10%	10%	10%
Pron → Noun	17%	17%	17%	17%	17%
Name → Pron	1%	1%	1%	1%	1%
Name → Name	18%	18%	18%	18%	18%
Name → Noun	3%	3%	3%	4%△	4%△
Noun → Pron	1%	1%	1%	1%	1%
Noun → Name	9%	9%	9%	9%	9%
Noun → Noun	21%	21%	21%	21%	21%
total num. errors	4844	**4800**	4853△	4861△	**4835**

Table 3: Recall Error Analysis of some of the best models per type of anaphor and antecedent. A decrease in the amount of errors is marked in **bold**, and an increase: with the symbol △.

problematic for the individual feature models.

4.2.2 Recall Error Analysis

The error analysis in terms of recall presented in Table 3 was not as informative as the precision one. Most of the systems perform similarly, with small variations in raw counts per error category.

Only two of the models manage to obtain lower total number of recall errors compared to the baseline: the embedding cluster with 44 fewer errors, and the combination model with 9. The overall good performance of the word embedding model in terms of precision is a result of a Precision-Recall trade-off.

5 Discussion

On the topic of selecting a word embedding model, we made several observations. In the automatic evaluation and error analysis for most of our experiments, including ones not reported here, the *glove* word embedding proved to be most useful. The *deps* model stood out in conjunction with the cosine similarity feature. Given that it has the smallest vocabulary of the three, it would be interesting so see how this type of embedding performs for the task when trained on a bigger data set. This would also allow us to fairly compare the former model, encoding more topical similarity, and the latter with more functional similarity. We believe that both types of knowledge are of importance with respect to coreference resolution.

The error analysis presented in subsection 4.2 suggests the need for a task-specific word embedding with a special handling of pronouns and names. While all tested feature configurations were beneficial for resolving coreferences involving common nouns, for several other categories involving the aforementioned types of expressions, they did not improve and sometimes even lead to a degradation in precision. The exploration of suitable task-specific models which address these issues is left to future work.

We observed a trade-off between performance in terms of F1 measure and computational cost. Our results partially support the findings of Yu et al. (2013) - the embedding cluster feature was the most cost-effective way to provide word embedding information to the coreference tool. However, it did not lead to better coreference resolution performance over the direct use of the word embedding. The word embedding feature did lead to longer training and testing, but using the PCA reduction technique improved on both running time and F1 measure.

Additional complexity is added to the coreference model when attempting to combine several of the word embedding features. Moreover, it is not trivial to find a good combination - selecting the features configurations which perform best in isolation did not necessarily lead to a more successful combination, as can be seen in Table 1.

We performed an additional study of the effect of the newly introduced word embedding features when some of the original features had been re-

Feature set	F1	ΔF1
baseline	59.19	
+EC$_{w2v,2.5k}$	59.56	+0.37
+WE$_{glove,pca\text{-}50}$	59.67	+0.48
-gender	59.10	-0.09
-gender, +EC$_{w2v,2.5k}$	59.18	-0.01
-gender, +WE$_{glove,pca\text{-}50}$	59.34	+0.15
-number	59.39	+0.20
-number, +EC$_{w2v,2.5k}$	59.33	+0.14
-number, +WE$_{glove,pca\text{-}50}$	59.43	+0.24
-head	59.19	+0.00
-head, +EC$_{w2v,2.5k}$	59.28	+0.09
-head, +WE$_{glove,pca\text{-}50}$	59.29	+0.10
-head_NE	59.14	-0.05
-head_NE, +EC$_{w2v,2.5k}$	59.34	+0.15
-head_NE, +WE$_{glove,pca\text{-}50}$	59.34	+0.15
-semclass	59.23	+0.04
-semclass, +EC$_{w2v,2.5k}$	59.44	+0.25
-semclass, +WE$_{glove,pca\text{-}50}$	59.46	+0.27
-finetype	55.50	-3.69
-finetype, +EC$_{w2v,2.5k}$	55.98	-3.21
-finetype, +WE$_{glove,pca\text{-}50}$	55.74	-3.45

Table 4: Interaction of some of the original and newly introduced features. ΔF1 denotes the difference in the performance of a model compared to the baseline.

moved. The goal of this experiment was to determine to what extend the knowledge encoded by these features overlaps with some of the semantic information already present in the original ones, and whether they might be better alternatives to them. These include: gender and number information, head word and Named Entity (NE) tag of the head word, and semantic class and fine type of the mention. Table 4 presents a summary of the results.

None of the new features seem to completely cover the information encoded by the original ones. Rather, they work best in the setting where they interact with each other. The dense word embedding feature includes more of the *number* and *gender* information, compared to the embedding cluster feature. An explanation for this could be that the different dimensions of an embedding of a word encode some of these properties. Thus the word embedding feature is a better way of providing them to the coreference system, as each dimension is kept as a separate numeric feature. This information is partially lost when performing clustering. For head word, its NE tag, and semantic class, we do not observe a big difference between the performance of the two new features. Both of them seem to successfully encode information on the semantic class of a word. For the fine type feature, the embedding cluster outperforms the word embedding feature, but neither manage to cover much of this information.

6 Conclusion

This work offers some insights on how word embeddings can be applied to the task of coreference resolution. We present three different features derived from word embeddings, and show that they influence the coreference resolution process in different ways. These include a setting in which each dimension of the embedding is a separate numeric feature, an embedding cluster which approximates a semantic class, and a set of cosine similarity features, which incorporate some contextual information.

Our evaluation results and error analysis show that each of these features helps to improve over the baseline coreference system's performance. We observed a reduction in the total number of precision errors. Moreover, all features lead to a reduction in the amount of precision errors in resolving references between common nouns. These results indicate that they successfully incorporate some lexical semantic and world knowledge into the process.

One of the ways in which we would like to extend the current experiments in future work is by creating a task-specific word embedding. Our error analysis shows that the current approach degraded the precision for the resolution of pronouns and names. For the former, word embeddings do not contribute much without the availability of more contextual information. We will investigate some specific examples to determine the reason for the latter.

We would further like to study the influence of models encoding more topical in comparison to ones with more functional similarity to the coreference resolution task.

References

David L. Bean and Ellen Riloff. 2004. Unsupervised learning of contextual role knowledge for coreference resolution. In *HLT-NAACL*. The Association for Computational Linguistics, pages 297–304.

Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. *CoRR* .

Hal Daumé III and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '05.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Revisiting embedding features for simple semi-supervised learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 110–120.

Otto Jespersen. 1949. *A Modern Grammar On Historical Principles. Part 7 Syntax*. Taylor and Francis Ltd, London, UK.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*. pages 28–34.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Sebastian Martschat, Patrick Claus, and Michael Strube. 2015. Plug latent structures and play coreference resolution. In *ACL*.

Sebastian Martschat and Michael Strube. 2014. Recall error analysis for coreference resolution. In *In proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Sebastian Martschat and Michael Strube. 2015. Latent structures for coreference resolution. In *ACL*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Curran Associates Inc., NIPS'13.

Dan Klein Mohit Bansal. 2012. Coreference semantics from web features. The Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543.

Altaf Rahman and Vincent Ng. 2011. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, HLT '11, pages 814–824.

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

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. *CoRR* .

Mo Yu, Tiejun Zhao, Daxiang Dong, Hao Tian, and Dianhai Yu. 2013. Compound embedding features for semi-supervised learning. In *HLT-NAACL*.