Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 27–33
Marseille, 11–16 May 2020

On the Influence of Coreference Resolution on Word Embeddings in Lexical-semantic Evaluation Tasks

Alexander Henlein, Alexander Mehler
Goethe University
Frankfurt am Main 60323, Germany
{henlein, mehler}@em.uni-frankfurt.de

Abstract

Coreference resolution (CR) aims to find all spans of a text that refer to the same entity. The F1-Scores on these task have been greatly improved by new developed End2End-approaches (Lee et al., 2017) and transformer networks (Joshi et al., 2019b). The inclusion of CR as a pre-processing step is expected to lead to improvements in downstream tasks. The paper examines this effect with respect to word embeddings. That is, we analyze the effects of CR on six different embedding methods and evaluate them in the context of seven lexical-semantic evaluation tasks and instantiation/hypernymy detection. Especially in the last task we hoped for a significant increase in performance. We show that all word embedding approaches do not benefit significantly from pronoun substitution. The measurable improvements are only marginal (around 0.5% in most test cases). We explain this result with the loss of contextual information, reduction of the relative occurrence of rare words and the lack of pronouns to be replaced.

Keywords: Word Embedding, Anaphora, Coreference Resolution

1. Introduction

Many NLP systems use word embeddings as a fast to learn resource that captures important lexical information (Mikolov et al., 2013). Once trained, embeddings can be used in many different tasks, like Coreference Resolution (Lee et al., 2018), Emotion Detection (Felbo et al., 2017), Biomedical Natural Language Processing (Wang et al., 2018), Image Caption Generation (Vinyals et al., 2015) or Text Classification (Uslu et al., 2019). Most of them rely on local information delimited by context windows or dependency parents to predict word relations (Levy and Goldberg, 2014). This approach encounters problems wherever semantic relationships have to be captured, which are expressed by coreference, as the following example illustrates:

> Edgar Allan Poe was an American writer.
> Poe is best known for his poetry.

Based on a context window-based approach of a maximum of five right neighbors, we get data to examine the relationship of *Poe* and *writer* and of *his* and *poetry*. But the model is not informed about a relationship between *Poe* and *poetry* when using a too small window. Obviously, the detour via the use of overly large window sizes (which would capture wanted as well as unwanted co-occurrences) can be prevented by a coreference resolution which replaces *his* with *Poe*.

The mapping of different linguistic expressions to the same entity is called *Coreference Resolution* (CR) (Ponzetto and Poesio, 2009). Previous systems were computationally very intensive and required a large NLP pipeline to calculate the required features (Clark and Manning, 2015; Wiseman et al., 2016; Clark and Manning, 2016; Poesio et al., 2016). The currently most modern system (Lee et al., 2018; Joshi et al., 2019a) does not need any of these features, therefore it is now possible to perform CR in a reasonable time. The resulting state-of-the-art score is 79.6% F1-Score for English. In this paper, we use CR as pre-processing step for training word embeddings, replace pronouns with their first mention, and evaluate the final word embeddings on different tasks. There are several approaches to evaluating word embeddings, which can be divided into *extrinsic* and *intrinsic* tasks. Extrinsic is the evaluation on downstream tasks such as POS tagging. Intrinsic evaluations explore word data about syntactic or semantic relations. The *Word Similarity* (WS) task, for example, evaluates how well the dot product of two word pairs correlates with the scores of human annotations (Jastrzebski et al., 2017). In this paper, we analyze the influence of resolving anaphoric relations on computing word embeddings by means of intrinsic approaches. As shown above, anaphoric relations are usually lost in training, although they manifest important relationships between words. Our experiments show that none of the embeddings analysed is improved by mention substitution – in any event, the improvements are only marginal. We explain this result with the loss of contextual information, reduction of the relative occurrence of rare words and the lack of pronouns to be replaced. The paper is organized as follows: Section 2 gives a short overview of word embeddings and of CR. Then we present our approach to enhancing word embeddings based on CR in Section 3. The experimental setup is described in Section 4. The results in Section 5. A prospect to future work is presented in Section 6.

2. Related Work

Pre-trained word embeddings (Mikolov et al., 2013; Ling et al., 2015; Pennington et al., 2014; Levy and Goldberg, 2014; Komninos and Manandhar, 2016) are a standard component of most modern NLP architectures. However, most of these systems are based only on local word information, such as skip-grams (e.g. Mikolov et al. (2013) or Ling et al. (2015)) or dependency relation-based windows (e.g. Levy and Goldberg (2014) or Komninos and Manandhar (2016)).

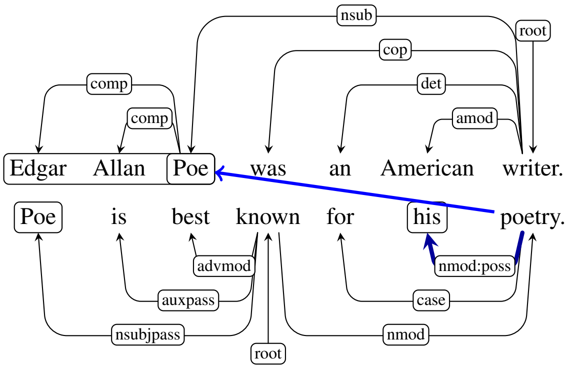

Figure 1: Dependency trees of two consecutive sentences. The blue arrow from *poetry* to *Poe* indicates the expanded context that is mediated by *his*.

Word	Context
poe	writer/nsub^{-1}, (poetry/nmod:poss^{-1})
writer	poe/nsub
his	poetry/nmod:poss^{-1}
poetry	his/nmod:poss, (poe/nmod:poss)

Table 1: Example input (reduced) of Levy and Goldberg (2014) based embeddings induced by the example of Figure 1. The additional contexts in parentheses are achieved with the help of CR.

Only recently, new systems have been introduced which are trained on large contexts using LSTMs (Peters et al., 2018) or large neural attention systems (Transformers) based on more complex transfer-learning tasks (Devlin et al., 2018; Liu et al., 2019) and are therefore not limited to local information – but at the price of additional computational complexity. At the same time, the list of proposals for new embedding methods that are pre-trained on ever larger corpora from more and more areas (genres, registers etc.) of more and more languages is constantly growing (Grave et al., 2018; Bojanowski et al., 2017; Radford et al., 2019). In recent years, the impact of various features such as POS-tags, subword information, semantic relations and in-domain data on word embeddings have been analyzed (Rezaeinia et al., 2017; Wendlandt et al., 2018; Bojanowski et al., 2017; Boleda et al., 2017; Gupta et al., 2017) and improved results have been obtained.

In this paper we complement this research and ask about the effects of CR on word embeddings. This is done by example of six methods of computing word embeddings: Cbow (Mikolov et al., 2013), Skip (Mikolov et al., 2013), Glove (Pennington et al., 2014), Wang (Ling et al., 2015), Levy (Levy and Goldberg, 2014) and Komninos (Komninos and Manandhar, 2016).

3. Coreference Substitutions for Enhancing Word Embeddings

In this section, we briefly introduce a formal apparatus to model coreference. Let

$$T = (w_1, \dots, w_i, \dots, w_n) \tag{1}$$

be a document with n tokens and words (lemmas) $w = L(w_i)$ at position i. To avoid grammatical issues (especially morphological ones), we lemmatize all tokens in T. A *mention*

$$m_{i:j} = (w_i, \dots, w_j) \tag{2}$$

is then defined as a continuous segment of tokens of T. Let

$$M = (m^1, \dots, m^p), p \leq n, \tag{3}$$

be the sequence of all mentions observed in T, sorted by occurrence. A mention m^i is said to be *antecedent* to a mention m^j if both are co-referent (and thus connected by a co-reference link) and if $i < j$. We denote this antecedence by $m^i < m^j$. Then we define the function

$$\text{first}(m^j) = \arg \min_{m^i \in \{m^i < m^j \mid i \in \{1,\dots,j-1\}\}} \{i\} \tag{4}$$

which returns the antecedent of m^j of lowest index and write $m^i \ll m^j \Leftrightarrow \text{first}(m^j) = m^i$.

3.1. Extending the informational scope of window-based embeddings

Our approach to extending window-based embeddings by means of CR is the following: For all pronominal mentions m^i, for which $\text{first}(m^i)$ is not pronominal, we replace:

$$m^i \leftarrow \text{first}(m^i) \tag{5}$$

This means that we replace each pronoun with its lowest index antecedent which in our case is represented by a corresponding lemma or multiword expression as shown in the following example:

. . . his poetry. ↦ *. . . Edgar Allan Poe poetry.*

So far, our replacement procedure only considers pronouns. The reason is that we expect the greatest loss of information from not replacing them. In this way, we avoid problems that we would get if we replaced phrasal mentions (e.g. more complex noun phrases) with their phrasal antecedents.

3.2. Extending the informational scope of dependency-based embeddings

For embeddings derived from dependency trees, we choose an approach that explores the underlying dependency relations. Let

$$D(w, T) = \{d(w_{i_1}), \dots, d(w_{i_k})\} \tag{6}$$

be the set of all parent tokens $d(w_{i_h})$ to which the tokens w_{i_h}, $h = 1..k$, of lemma $w = L(w_{i_h})$ are directly dependent in text T. Conversely,

$$D^{-1}(w, T) = \{w_i \in T \mid L(d(w_i)) = w\} \tag{7}$$

is the set of all tokens that directly depend on some token of lemma w in T. A tabular representation of these sets derived from the text sample of Figure 1 is shown in Table 1. The procedure for extending the informational basis for computing dependency-based embeddings is now as follows: for each lemma for which there is a token

Type	WS	Average				MEN				WS353				SimLex999				RW				MTurk-287				Google				SemEval2012_2			
		c	p	h	ph	c	p	h	ph	c	p	h	ph	c	p	h	ph	c	p	h	ph	c	p	h	ph	c	p	h	ph	c	p	h	ph
Cbow	2	42.29	42.11	**43.37**	42.43	67.54	67.02	**68.56**	67.38	53.87	54.56	**55.28**	54.32	**35.42**	34.75	35.09	34.97	25.28	25.31	**25.98**	24.42	61.63	60.76	**61.75**	61.23	33.68	34.61	**39.20**	36.15	**18.59**	17.77	17.70	18.52
Cbow	5	44.64	44.47	**45.80**	44.94	70.45	70.21	**71.24**	70.71	58.24	57.42	**58.61**	58.52	37.42	37.18	**37.78**	37.57	24.36	24.32	**24.78**	24.35	61.60	60.64	**62.21**	61.89	41.63	43.08	**46.37**	43.94	18.79	18.45	**19.61**	17.62
Cbow	10	45.68	45.48	**46.37**	46.07	72.25	71.93	**72.60**	72.45	60.70	60.31	60.69	**60.72**	37.32	37.15	**37.61**	37.37	**25.28**	24.70	24.83	24.13	62.24	62.63	63.80	**63.88**	44.41	45.65	**46.97**	46.20	17.57	16.00	**18.10**	17.73
Skip	2	47.58	47.71	**48.59**	48.13	**73.54**	73.43	74.35	73.50	66.03	65.57	**67.76**	66.47	40.92	40.84	41.25	**41.81**	31.33	31.46	31.95	**32.12**	61.66	61.88	**62.96**	61.48	41.53	42.12	**42.87**	41.94	18.02	18.71	19.03	**19.59**
Skip	5	49.61	49.15	49.32	**49.63**	75.44	75.64	**75.83**	75.55	**69.09**	68.80	67.98	68.95	**40.44**	39.69	40.10	40.35	**32.43**	30.96	31.50	32.04	63.92	63.43	**65.34**	65.30	**48.33**	47.96	47.90	48.19	**17.63**	17.58	16.56	17.06
Skip	10	**48.92**	48.61	48.64	48.61	76.19	76.25	**76.28**	76.05	68.14	68.12	**68.38**	68.08	38.07	**38.24**	38.03	38.13	**30.33**	28.37	29.48	29.12	65.89	65.91	65.70	**66.04**	48.31	**48.59**	48.40	48.52	**15.50**	14.82	14.20	14.31
Glove	2	33.94	32.96	**34.20**	34.00	62.17	61.00	**62.95**	62.29	43.20	40.73	**43.37**	41.23	27.47	27.26	**28.01**	27.55	14.04	13.51	13.99	**14.05**	**51.92**	50.02	50.84	50.90	25.28	25.15	25.78	**25.98**	13.47	13.05	14.47	**15.99**
Glove	5	38.29	37.28	38.02	**38.41**	68.47	66.54	68.45	**68.84**	**47.51**	45.78	47.10	46.93	**29.35**	27.18	28.39	28.94	16.56	16.39	16.18	**16.64**	53.52	53.79	**54.18**	54.11	37.99	36.93	**38.04**	37.87	14.61	14.40	13.77	**15.54**
Glove	10	**39.43**	38.23	39.06	39.16	70.00	68.16	69.47	69.42	48.04	46.62	47.48	47.38	29.06	27.04	27.86	28.32	16.87	16.53	16.40	16.71	55.14	54.66	55.07	55.57	42.42	41.89	42.42	42.22	14.50	12.74	14.74	14.48
Wang	2	47.26	47.36	**47.96**	47.46	72.03	71.63	**73.34**	71.78	65.45	66.74	**67.65**	66.32	43.22	42.94	**43.30**	41.99	**33.36**	32.90	33.27	33.34	**59.63**	59.37	59.32	**59.63**	36.89	38.54	**38.69**	38.22	20.22	19.41	20.13	**20.93**
Wang	5	**48.37**	48.04	48.28	48.21	73.03	73.30	**73.74**	73.48	68.25	67.33	**68.64**	67.92	41.33	41.02	41.71	**42.28**	32.68	31.34	32.59	**33.05**	**62.39**	60.38	59.93	59.15	42.38	**43.78**	43.01	42.83	18.56	**19.12**	18.36	18.75
Wang	10	47.56	48.50	48.38	**48.58**	73.06	73.89	**74.17**	73.60	68.10	**68.96**	68.93	68.79	**41.70**	41.32	41.21	41.58	32.14	31.61	31.87	**32.47**	56.89	**60.49**	58.04	60.28	44.02	45.16	**45.59**	45.02	16.99	18.05	**18.82**	18.31
Levy		41.80	-	-	**41.97**	66.54	-	-	**66.95**	60.59	-	-	**61.76**	46.16	-	-	**46.40**	**31.64**	-	-	**31.64**	54.35	-	-	**54.70**	12.21	-	-	**12.49**	**21.11**	-	-	19.86
Komninos		**47.45**	-	-	47.26	**72.68**	-	-	72.50	**62.84**	-	-	62.68	**42.09**	-	-	41.29	**33.73**	-	-	33.46	**61.00**	-	-	60.80	38.84	-	-	**40.28**	**20.97**	-	-	19.78

Table 2: Evaluation of different embedding types with different window sizes. c stands for the original dataset, p where we replaced only pronouns, h where we only replaced every mention with the mention head and ph, where we replaced only pronouns with the corresponding antecedent.

that directly dominates a pronominal anaphoric mention, we add a dependency link from this token to the non-pronominal antecedent of lowest index of this pronoun. If this antecedent consists of several tokens, the root node of the corresponding dependency subtree is used as the target of the link. More formally: for each anaphoric pronoun $w_k \in D^{-1}(w, T)$ depending on token $d(w_k)$ of lemma $w = L(d(w_k))$ such that there exists a mention $w_k = m^j \in M$ (pronominal mentions are one-place), we extend the set of dependents $D^{-1}(w, T)$ of w as follows:

$$\begin{aligned} \dot{D}^{-1}(w,T) \;\; = \;\; & D^{-1}(w,T) \cup \\ & \{r(\text{tree}(m^i)) \mid \exists m^j \in M \\ & \exists w_k \in D^{-1}(w,T){:} \\ & w_k = m^j \wedge m^i \ll m^j\} \end{aligned} \tag{8}$$

where $r(\text{tree}(m^i))$ denotes the root of the dependency subtree $\text{tree}(m^i)$ spanned by mention m^i. A dependency tree showing an added link between *poetry* and *Poe* is exemplified in Figure 1. The corresponding extended contexts are indicated by brackets in Table 1. By analogy to $D^{-1}(w, T)$, we extend $D(w, T)$, so that added links can be processed in both directions by means of the approach of Levy and Goldberg (2014). Note that we only consider anaphoric, but not cataphoric references which also allow for adding dependency links.

4. Experiments

4.1. Data Sets and Models

Our dataset used for training consists of the first paragraphs of 1.000.000 Wikipedia articles (effects of smaller datasets are analysed in section 5.4) with almost 300 millions tokens, of which over 4 million (of almost 5.5 million) pronouns have been replaced or extended. The models used are the Skip and Cbow variant of Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014) and Wang2Vec (Ling et al., 2015), Levy (Levy and Goldberg, 2014) and Komninos (Komninos and Manandhar, 2016). Word2Vec, Glove and Wang were trained with a fixed vocabulary of the 400.000 most commonly lemmatized tokens and Levy and Komninos with all lemmatized tokens that occurred at least 15 times in the data set. We trained all embeddings with a size of 300, standard parameters, window sizes of 2, 5 and 10, and 25 iterations.

4.2. Pre-processing

We used Spanbert-Base of Joshi et al. (2019a) for coreference resolution. For the needed dependency features we used the AllenNLP's (Gardner et al., 2018) implementation of Dozat and Manning (2016). For tokenization, lemmatization and POS tags, Spacy (Honnibal and Montani, 2017) was used.

5. Evaluation

5.1. Word Similarity

The first analyses on the generated word vectors ran over various word similarity tasks. All results are listed in table 2. For evaluation, we used the benchmark tool of Jastrzebski et al. (2017)[1] as it computes the accuracy for a lot of important Word Similarity and Analogy Tasks. We used: (MEN (Bruni et al., 2014), WS353 (Finkelstein et al., 2002), SimLex999 (Hill et al., 2015), RW (Luong et al., 2013), MTurk-287 (Radinsky et al., 2011), Google (Mikolov et al., 2013), SemEval2012_2 (Jurgens et al., 2012)). We compare the unmodified dataset (c-version) with a version, were we replaced pronouns with the complete antecedent (p-version, described in section 3.1), replaced everything with the mention-head (h), and replaced only pronouns with the mention-head (ph-version, described for dependency in section 3.2). For most context window-based embeddings, the results based on the data set containing the co-reference do not differ markedly. It is noteworthy that the p-version is usually worse than the c-version. The observed reductions in the case of context window-based approaches can be explained by the effect of the loss of semantic contexts (see section 5.5). The h- and ph-versions perform therefore better. We therefore only consider these versions in further analyses. But still, some embeddings have a tendency towards slightly better results (e.g. Cbow), while others tend to get a little worse (Wang2Vec). The best responding test data is by far Google, with an increasing of 5.52% with Cbow (2). The worst results were obtained on the RW and MTurk-287 data set. Intuitively, the results for coreference embeddings are better for small window sizes.

Ins/Hyp (Window)	Conc		Diff		DDSq	
	c	ph	c	ph	c	ph
I-NotInst (10)	81.09	81.09	78.97	80.48	80.73	**82.00**
I-Inverse (10)	98.79	98.34	99.09	99.24	**98.79**	99.24
I-I2I (10)	95.80	**96.40**	92.20	92.80	92.20	92.80
I-Union (10)	**84.94**	84.41	77.58	76.88	79.77	78.98
H-NotHyp (10)	55.16	55.53	54.32	54.23	72.84	**73.12**
H-Inverse (10)	81.75	79.56	**83.84**	82.01	83.76	81.92
H-C2C (10)	69.03	68.57	64.15	64.52	79.23	**79.78**
H-Union (10)	42.73	42.24	40.79	40.30	**52.98**	52.26
I-Union (2)	**86.25**	85.20	77.67	77.50	79.16	78.37
H-Union (2)	45.13	44.36	43.19	41.29	**53.61**	53.29

Table 3: Results on the Instances and Concepts dataset (Boleda et al., 2017) with the Cbow model.

Ins/Hyp	**Conc**		**Diff**		**DDSq**	
	c	ph	c	ph	c	ph
I-NotInst	**82.96**	80.84	81.45	80.54	81.90	80.84
I-Inverse	99.55	99.70	99.70	**99.85**	99.70	**99.85**
I-I2I	98.40	**98.60**	96.01	95.61	96.01	95.61
I-Union	**87.16**	87.07	80.00	80.17	81.22	80.70
H-NotHyp	56.55	57.66	53.67	53.11	67.69	**68.52**
H-Inverse	84.37	84.02	85.76	**86.46**	85.59	86.11
H-C2C	72.45	72.54	66.30	66.02	**74.56**	73.92
H-Union	44.74	45.06	41.13	41.08	**50.25**	50.16

Table 4: Results on the Instances and Concepts dataset (Boleda et al., 2017) with the Levy.

5.2. Instances versus Concepts

Next, we tested whether the vectors could better distinguish between *instances* or *concepts*. The embedding task including the test dataset was presented by Boleda et al. (2017). The data set consists of word pairs (x, y) where a linear classifier is used to decide whether x is an instance or a hyponym of y. As a negative example, the data set contains various error cases, like swap(x, y) (inverse). Further details can be found in Boleda et al. (2017). As in the original work, we trained a linear logistic regression classifier with the concatenation (Conc), the difference (Diff) and the squared difference (DDSq) of the vectors as input. We used scikit-learn (Pedregosa et al., 2011) for implementing this. The results for the Cbow model are listed in table 3 and for the Levy model in table 4. Again, the vectors do not seem to achieve any performance improvement. However, with regard to the Union dataset, it appears that the results have tended to get worse.

5.3. Feature Analysis

To analyze the results, we took the classification results of the development and test dataset from the linear classifier of section 5.2 to decide, which words where classified better or worse. With this information we trained a *Decision Tree* (DT) and a Support Vector Machine (SVM) to predict whether the classification of a word w is improved or worsened when taking into account the following features: 1. How often did we use w to replace a pronoun according to Section 3 (Replacer), 2. Log-frequency of w in the corpus (VocabC), 3. Frequency in the test set (inTest) and 4. Character count of w (WordLen). The generated DT for the Cbow model with window size 10 on the H-Union dataset is shown in Figure 2. One observation is that words that appear more frequently in the corpus become slightly better, whereas words that are already rare tend to get worse. But as soon as words occur too often, they tend to get worse again. It seems that the embeddings already contain all necessary neighborhood information in the case of high-frequency words. Rare words, on the other hand, become even rarer and therefore their vector representations are worsened. The strongest feature for the SVM was VocabC and the log of Replacer, so we trained a small version with only these two features to show their behaviour in a two-dimensional space (see Figure 3). The results are similar to those of DT. However, with the decision boundaries it is recognizable how the word frequencies correlate with the results. The words tend to get better if they are neither too frequent nor too rare in the training data. The same applies to the replacement. One possible explanation is that common words already cover all information. Rare words, on the other hand, are rarely referenced by anaphora and do not benefit from this procedure. It should be noted that this is not so easy to detect with smaller window sizes.

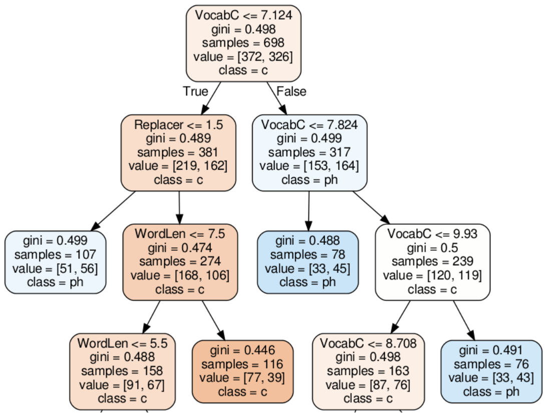

Figure 2: Decision Tree for classifying the error distribution on the H-Union dataset. Red nodes stand for word embeddings, which tend to get worse through pronoun substitution. Blue nodes tend to get better through pronoun substitution. Gini is a measure of the probability that a randomly selected element from the data will be misclassified. Value stands for the division of the samples into the two classes at this node.

5.4. Corpus Size

We have also tested different corpus sizes, but have not found any significant effect for them either. The results are listed in Table 5. Doc Count is the randomly selected number of Wikipedia articles.

[1] https://github.com/kudkudak/word-embeddings-benchmarks

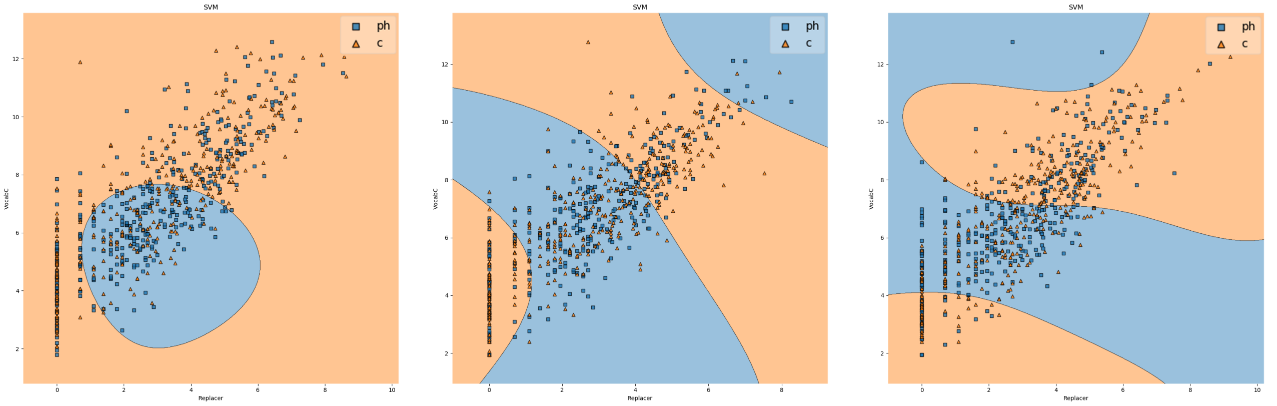

Figure 3: Error distribution of the SVM with log(Replacer) on the x-axis and VocabC on the y-axis. Cbow (left), Skip (middle), Glove (right) with window size 10. Red areas stand for word embeddings, which tend to get worse through pronoun substitution. Blue areas tend to get better through pronoun substitution. The decision-boundaries reveal, that words that are neither too frequent nor too rare in the corpus tend to produce better results if they are neither replaced too often nor too rarely.

Doc Count	**Average**			
	c	p	h	ph
100	4.41	5.05	5.00	**5.15**
1000	14.92	15.54	15.35	**16.13**
10000	**29.87**	29.79	28.80	29.67
100000	38.56	38.69	**39.54**	39.01
1000000	43.35	43.54	**43.71**	43.58

Table 5: Average results (see section: 5.1) for Cbow with vector size 100 and window size 10 on different amounts of Wikipedia articles.

5.5. Explanation of the results

5.5.1. Loss of Semantic Contexts

Windows-based embeddings achieve their quality by looking at which words appear together in observed windows. In the example

> *[Edgar Allan Poe]*$_1$ *was an American writer. [Poe]*$_1$ *is best known for [his]*$_1$ *poetry.* ↦ *[Edgar Allan Poe]*$_1$ *was an American writer. [Poe]*$_1$ *is best known for [Edgar Allan Poe]*$_1$ *poetry.*

the distance between associated words (e.g. *writer* and *poetry*) increases so much by the substitution of *his* that the system is no longer informed about their association in this example. This effect is increased by the fact that we always replace pronouns with possibly longer mentions (experiment p). In this way, we amplify the effect that we originally wanted to avoid. The example also shows that substituting pronouns is not a trivial task and can distort the semantics of a sentence. The same may happen with syntax as shown in the example above.

5.5.2. Word Frequency

We were able to show that words that are neither too frequent nor too rare in the corpus tend to produce better results if they are neither replaced too often nor too rarely. In contrast, the use of frequent words to replace pronouns tend to noise out their already well-documented contextual information within the original corpus. And for rare words, the additional context information gained by CR is not detailed enough to calculate better embeddings for them. However, it should be noted that the replacements have only led to a minimal increase in the volume of data.

5.6. Discussion

Our goal was not primarily to achieve the best results for the evaluation tasks we carried out, but to investigate the effects of coreference resolution on computing word embeddings. Actually, there is an effect, but only a small one. This finding indicates the need to further elaborate the interplay of pre-processing routines like coreference resolution and downstream tasks such as training word embeddings. With a more elaborated substitution function $\text{first} : M \rightarrow M$ than the one implemented here better results might be achieved. An extension would be, for example, training with both sentences, the ones in which substitutions are made and the original ones. Replacing with (parts of) nominal phrases might distort the training as well. The use of only named entities could help with this problem, but would further reduce the amount of information obtained.

6. Conclusion

We experimented with improving word embeddings based on CR as a pre-processing step. We have shown that word embedding approaches do not tend to benefit significantly from pronoun substitution. The measurable improvements were only marginal, even though we could achieve strong improvements with Cbow on the Google dataset. In future work, we want to analyze the effect of linking all mentions of the same reference chain with each other (completely connected graph). In addition, we want to find out which dependency edges contribute to the information gain by training corresponding classifiers.

7. Bibliographical References

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Boleda, G., Gupta, A., and Padó, S. (2017). Instances and concepts in distributional space. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 79–85.

Bruni, E., Tran, N.-K., and Baroni, M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Clark, K. and Manning, C. D. (2015). Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1405–1415.

Clark, K. and Manning, C. D. (2016). Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dozat, T. and Manning, C. D. (2016). Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., and Zettlemoyer, L. (2018). Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Gupta, A., Boleda, G., and Padó, S. (2017). Distributed prediction of relations for entities: The easy, the difficult, and the impossible. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 104–109.

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

Honnibal, M. and Montani, I. (2017). spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7.

Jastrzebski, S., Leśniak, D., and Czarnecki, W. M. (2017). How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. *arXiv preprint arXiv:1702.02170*.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2019a). SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Joshi, M., Levy, O., Weld, D. S., and Zettlemoyer, L. (2019b). BERT for coreference resolution: Baselines and analysis. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Jurgens, D. A., Turney, P. D., Mohammad, S. M., and Holyoak, K. J. (2012). Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 356–364. Association for Computational Linguistics.

Komninos, A. and Manandhar, S. (2016). Dependency based embeddings for sentence classification tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1490–1500.

Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (2017). End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.

Lee, K., He, L., and Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392*.

Levy, O. and Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 302–308.

Ling, W., Dyer, C., Black, A. W., and Trancoso, I. (2015). Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304.

Liu, X., He, P., Chen, W., and Gao, J. (2019). Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Luong, T., Socher, R., and Manning, C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of NAACL-HLT 2018*, pages 2227–2237.

Poesio, M., Stuckardt, R., and Versley, Y. (2016). *Anaphora resolution*. Springer.

Ponzetto, S. P. and Poesio, M. (2009). State-of-the-art nlp approaches to coreference resolution: Theory and practical recipes. In *Tutorial Abstracts of ACL-IJCNLP 2009*, pages 6–6. Association for Computational Linguistics.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

Radinsky, K., Agichtein, E., Gabrilovich, E., and Markovitch, S. (2011). A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM.

Rezaeinia, S. M., Ghodsi, A., and Rahmani, R. (2017). Improving the accuracy of pre-trained word embeddings for sentiment analysis. *arXiv preprint arXiv:1711.08609*.

Uslu, T., Mehler, A., and Baumartz, D. (2019). Computing Classifier-based Embeddings with the Help of text2ddc. In *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing 2019. accepted.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., and Liu, H. (2018). A comparison of word embeddings for the biomedical natural language processing. *Journal of biomedical informatics*, 87:12–20.

Wendlandt, L., Kummerfeld, J. K., and Mihalcea, R. (2018). Factors influencing the surprising instability of word embeddings. *arXiv preprint arXiv:1804.09692*.

Wiseman, S., Rush, A. M., and Shieber, S. M. (2016). Learning global features for coreference resolution. *arXiv preprint arXiv:1604.03035*.