Fuzzy paraphrases in learning word representations with a corpus and a lexicon

Anonymous Author(s)
Affiliation
Address
email

Abstract

We figure out a trap that is not carefully addressed in the previous works using lexicons or ontologies to train or improve distributed word representations: For polysemantic words and utterances changing meaning in different contexts, their paraphrases or related entities in a lexicon or an ontology are unreliable and sometimes deteriorate the learning of word representations. Thus, we propose an approach to address the problem that considers each paraphrase of a word in a lexicon not fully a paraphrase, but a fuzzy member (i.e., fuzzy paraphrase) in the paraphrase set whose degree of truth (i.e., membership) depends on the contexts. Then we propose an efficient method to use the fuzzy paraphrases to learn word embeddings. We approximately estimate the local membership of paraphrases, and train word embeddings using a lexicon jointly by replacing the words in the contexts with their paraphrases randomly subject to the membership of each paraphrase. The experimental results show that our method is efficient, overcomes the weakness of the previous related works in extracting semantic information and outperforms the previous works of learning word representations using lexicons.

1 Introduction

There have been many works and models to estimate the distributed representations of words, i.e. the word embeddings for a corpus [1, 18, 19, 4, 8, 14, 16, 15, 17, 23, 2]. Benefiting from the works, high quality word embeddings can be estimated efficiently nowadays.

Word embeddings are reported useful and improve the performance of the machine learning algorithms for many natural language processing tasks such as name entity recognition and chunking [26], text classification [24, 11, 10, 9], topic extraction [5, 12], and machine translation [29, 25].

Nevertheless, there is still room for improvement. For example, for the fine-grained sentiment analysis tasks such as predicating the number of stars of a review, the reported accuracy is much lower than the other text classification tasks [9]. It indicates the needs of word representations that embed the semantic information more efficiently.

Bojanowski et al. [2] attempt to improve word embeddings by involving character level information. There is a big improvement on syntactic questions in the word analogical reasoning task introduced by Mikolov et al. [15]. However, the accuracy for the semantic part is not improved in the reported results.

Some works [28, 27, 6, 3] try to estimate better word embeddings by using a lexicon or an ontology. The idea is simple: because a lexicon or an ontology contains well-defined relations about words, word embeddings of high quality can be learned from it, or we can refine trained word embeddings using the lexicon or the ontology.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

However, a problem is not well addressed in the previous works using lexicons to learn word embeddings: For a polysemantic word or utterance, its paraphrase in a lexicon or an ontology is not always its paraphrase in different contexts. For example, we can replace the word "Earth" in the sentence "Earth goes around the sun" with its paraphrase "Terra", however the same word "Earth" in the sentence "I fill the hole with earth" cannot be replaced with "Terra". Henceforth, the lexicon or the ontology is sometimes unreliable and deteriorates the learning of word embeddings for the polysemantic words and utterances.

In this paper, we propose a method to learn word embeddings using both a corpus and a lexicon that is able to alleviate the bad effect of polysemants, by estimating the degree of truth of each paraphrase in the lexicon. Our method for estimating is simple, efficient and easy to be combined with the previous learning algorithms on the basis of co-occurrences of words. The experimental results show that our method is efficient and outperforms the previous works.

2 Related works

2.1 Works on Learning Word Embeddings for a corpus

The first approaches learning word embeddings use n-gram model [1, 4, 8] and recurrent neural networks [14]. Recently, more efficient methods like continuous bag-of-words (CBOW) model and skip-gram (SG) model [16, 15], also called word2vec, provide a more efficient way to learn word embeddings based on the local co-occurrences of words in a corpus. Continuous bag-of-words tries to maximize the log probability of a word given its context, while skip-gram tries to maximize the log probability of the words in the context given a word. Negative sampling is an efficient algorithm to train them, that approximately maximizes the log probability for the targets by doing a logistic regression to discriminate the target word from noise randomly drawn from a noise distribution.

Bojanowski et al. [2] extend word2vec using character-level information. They achieve a considerable improvement for rare words and the syntactic part of the word analogical reasoning task [15]. However, they fail to improve the performance for the semantic part of the task.

Other works attempt to train word embeddings via global information [8, 23] and report improvement than the other works that use only local information. However the global information is still limited by the corpus. Some other works follow another approach that uses a lexicon or an ontology to improve the word embeddings.

2.2 Works on Learning Word Embeddings using Lexicons

The models proposed by Yu & Dredze [28], and Bollegala et al. [3] jointly learn word embeddings from a corpus and a semantic lexicon. The method proposed by Yu & Dredze [28] called jointRCM improves word representations by maximizing the similarity of the word representations of the paraphrase pairs in the lexicon jointly with word2vec. The works by Bollegala et al. [3] also improve the word embeddings by minimizing the distance of the word representations of related words. But they use not only the synonyms, but also the global information in corpus and other relationships such as antonyms and hypernyms.

Xu et al. [27] propose models called R-Net and C-Net. R-Net, for a triplet of words $(head, relation, tail)$ in a knowledge graph, minimizes the distance of the embedding vector of the tail word, from the sum of the vectors of the head word and the relation. On the other hand, C-Net makes a word less similar than the other words share the same category if the size of the category is large. They jointly train the R-Net and C-Net with skip-gram.

Faruqui et al. [6] concentrate on refining pre-trained word embeddings using semantic lexicons. However, as also pointed out by Bollegala et al. [3] as incompatibilities between the corpus and the lexicon, some features extracted from the corpus that are not contained in the corpus, such as those from idioms or new words not contained in the lexicon, are improperly removed.

2.3 A not Well Addressed Trap in Learning Word Representations using Lexicons

In most of the previous works using lexicons to learn word representations, weight coefficients are used to control the input from the lexicon. These coefficients are manually optimized with a separated

dataset. However, it cannot address the problem that the reliability of the paraphrases in a lexicon depends on different words and different contexts because a word or an utterance can have several meanings. For example, though "Terra" is the paraphrase of "Earth" in "Earth goes around the sun", but obviously not its paraphrase in "I fill the hole with earth".

3 The proposed method

3.1 Learning Word Embeddings with fuzzy paraphrases

Our method is based on two ideas. The first, if the meanings of word j and word k are ideally the same, they can replace each other in a text without changing the meaning and all the other implicit features of the text. Further, when we train word embeddings by predicating a word i by the context containing word j like word2vec [16, 15], the probability of word i keeps unchanged in that case even though word j is replaced by word k. Henceforth, we can learn word embeddings using both a corpus and a lexicon by learning the original texts and those that some words are replaced with their paraphrases at the same time in the ideal case.

The second, as described in section 1 and 2, the paraphrases of a word in a lexicon are not always the paraphrases in a certain text but depend on the contexts because of polysemantic words and utterances. Henceforth, if we simply consider all the paraphrases of a word in the lexicon fully the paraphrases for the whole corpus, they deteriorate the word embeddings of some words and texts. To avoid that, we consider each paraphrase of a word in the lexicon not fully included in the paraphrase set, but a fuzzy member with a grade of membership (i.e. a degree of truth). Then we reject some paraphrases for some contexts subject to their membership.

./figures/nnview.eps

Figure 1: Architecture of the proposed model.

For a text T, denote w_i the ith word in T, c the context window, w_j a word in the context window, L_{w_j} the paraphrase set of w_j in the lexicon L, w_k the kth fuzzy paraphrase in L_{w_j}, and x_{jk} the membership of w_k for w_j, based on the CBOW model [15] and the two ideas, we propose a model called continuous bag-of-fuzzy-paraphrases (CBOFP) to train word embeddings using both a corpus and a lexicon, by maximizing not only the probability of a word for a given context, but also the probability after some of the words in the context are replaced by their paraphrases randomly subject to a function of the membership of each paraphrase:

$$\sum_{w_i \in T}^{T} \sum_{(i-c) \le j \le (i+c)} \left[\log p(w_i|w_j) + \sum_{w_k \in L_{w_j}}^{L_{w_j}} f(x_{jk}) \log p(w_i|w_k) \right] \tag{1}$$

The function $f(x_{jk})$ of the membership x_{jk} returns 1 or 0 for different paraphrases of different contexts and reduces the probabilities of the bad replacements that deteriorate the word embeddings by returning 0 more for the paraphrases that have lower grades of membership. The model can be considered as a revised CBOW model with an additional layer whose output is weighted by $f(x_{jk})$ as shown in Figure 1.

3.2 Membership Estimation

If we want the control function $f(x_{jk})$ to reject bad replacements perfectly, $f(x_{jk})$ or the membership x_{jk} should consider all of the contexts because the similarity of the paraphrases depends on not only themselves but also the other contexts. However, it is not easy to train such a function.

Looking for a control function that is easy to train, we notice that if two words are more often to be translated to the same word in another language, the replacement of them are less likely to change the meaning of the original sentence. Thus, we use a function of the bilingual similarity (denoted as S_{jk}) as the membership function without considering the other contexts:

$$x_{jk} = g(S_{jk}) \tag{2}$$

Table 1: Different types of relationships of paraphrases in PPDB2.0[22, 21]

Relationship	Description
Equivalence	X is the same as Y
Forward Entailment	X is more specific than/is a type of Y
Reverse Entailment	X is more general than/encompasses Y
Exclusion	X is the opposite of Y / X is mutually exclusive with Y
OtherRelated	X is related in some other way to Y
Independent	X is not related to Y

There have been works about calculating the similarity of words using such bilingual information and a lexicon called the paraphrase database (PPDB) provides scores of the similarity of paraphrases [7, 22, 21] on the basis of bilingual features. We scale the similarity score of the paraphrase w_k to $[0, 1]$ in PPDB2.0 as the membership, and draw the values of $f(x_{jk})$ from a Bernoulli distribution subject to the membership calculated in this way. Denote S_{jk} the similarity score of word w_j and w_k in PPDB2.0, the value of $f(x_{jk})$ is drawn from the Bernoulli distribution:

$$f(x_{jk}) \sim Bernoulli(x_{jk}) \tag{3}$$

$$x_{jk} = \frac{S_{jk}}{\max\limits_{j \in T, k \in L} S_{jk}} \tag{4}$$

We find the control function defined above is efficient in the experiments as described later in section 4.

3.3 Training

Hence we do not need to train $f(x_{jk})$ using the method described above. The model can be trained by negative sampling proposed by Mikolov et al. [16]: For word w_O and a word w_I in its context, denote A_I as the set of the paraphrases for w_I accepted by $f(x_{jk})$, we maximize $\log p(w_O|w_I)$ by distinguishing the noise words from a noise distribution $P_n(w)$ from w_O and its accepted paraphrases in A_I by logistic regression:

$$\log p(w_O|w_I) = \log \sigma({v_{w_O}}' v_{w_I}) + \sum_{i=1}^{n} E_{w_i} \sim P_n(w)[\log \sigma(-{v_{w_i}}' v_{w_I})], w_i \neq w_O, w_i \notin A_I \tag{5}$$

Here, n is the number of total negative samples. $\sigma(x)$ is a sigmoid function, $\sigma(x) = 1/(1 + e^{-x})$.

3.4 Different types of paraphrases and the paraphrase set for each word

In PPDB2.0, there are 6 relationships for paraphrases on the basis of the thesis of MacCartney [13]. For word X and word Y, the different relationships of them defined in PPDB2.0 are shown in Table 1. We see that they are far more than we need as some of them are not the conventional "paraphrases" that can replace each other. Only the paraphrases of equivalence, forward entailment and reverse entailment are used in our method. For each word in the vocabulary, the paraphrases equal to it or entailed by it are put into its paraphrase set for learning. For example, denote each paraphrase in PPDB2.0 as $(headword, tailword, relationship)$, for word A, B, C, D, E, if there are paraphrases $(A, B, Equivalence)$, $(A, C, ForwardEntailment)$, $(D, A, ReverseEntailment)$, $(A, E, Independent)$, the paraphrase set for A is (B, C, D), and E is discarded.

4 Experiments and Results

4.1 The Corpus, Lexicon and Parameters

In the experiments, we used enwiki9[1] as the corpus to train our model. It contains the first one billion bytes in the English Wikipedia. After removing the meta-data, tags, hyperlinks, references, URL encoded characters and converting uppercase letters, spaces and spell digits, the corpus contains 123,353,508 tokens. Among them, there are 218,317 different words.

PPDB2.0 [22, 21] is used as the lexicon. It contains more than 100 million paraphrases and 26 thousand manually rated phrase pairs. Only the paraphrase pairs whose relationships are equivalence, forward entailment, or reverse entailment are used for our method in the experiments as described in 3.4.

200-dimension word embeddings are trained using our method for the experiments. The context window is set to 8, the number of negative samples is set to 25, the total number of iterations is set to 15 for training. We made an implementation of the proposed method learning enwiki9 and using PPDB2.0 as the lexicon available online[2].

4.2 Baselines

As baselines to compare with our proposed method, we use word2vec [16, 15] (Marked as CBOW and SG, for continuous bag-of-word and skip-gram, respectively), word2vec enriched with subword information [2] (Marked as Enriched CBOW and Enriched SG), GloVe [23], which are widely used to extract word embeddings from a corpus. We also compare our method with the other works using a lexicon to improve word embeddings, which are jointRCM [28], jointReps [3], RC-Net [27], and the method to retrofit pre-trained word embeddings using a lexicon (Marked as Retro) [6].

We used the public online available source code of word2vec[3], word2vec enriched with subword information[4], GloVe[5], jointRCM[6], jointReps[7],and Retro[8] to build them for the experiments. But for jointRCM and jointReps, the results in our experiments are one percent the number of the reported results in the papers . It is unreasonably low, even though the corpus in our experiments is smaller than those used in their experiments. Thus we use the reported results in the papers to compare with our method. The reported results of jointRCM are achieved using the New York Time 1994-97 subset from Gigaword v5.0 [20] containing 518,103,942 tokens. The reported results of jointReps are achieved with ukWaC[9] containing 2 billion tokens. For RC-Net, there are no publicly available implementations unfortunately. We report the published results in their paper that are also achieved with enwiki9 to compare with ours. For the other baselines, 200-dimension word embeddings are trained using enwiki9. The context window is set to 8. For word2vec and that enriched with subword information, the number of negative samples is set to 25. The word embeddings trained by CBOW and SG are both used for Retro respectively, and marked as Retro (CBOW) and Retro (SG).

4.3 Word Analogical Reasoning Task

The word analogical reasoning task is introduced by Mikolov et al. [15]. Given a quaternion of words (w_A, w_B, w_C, w_D) that w_A and w_B have the similar relationship with that of w_C and w_D, the objective is to predict w_D on the basis of w_A, w_B and w_C. Given the word embedding v_A, v_B and v_C for w_A, w_B and w_C, it can be solved by finding the word whose word embedding is the closest to $v_B - v_A + v_C$. The dataset is separated into two parts: the semantic part and the syntactic part. The semantic part is about analogical reasoning via semantic relationships, such as predicting the capital

[1] http://mattmahoney.net/dc/enwiki9.zip
[2] https://github.com/huajianjiu/Bernoulli-CBOFP
[3] https://code.google.com/archive/p/word2vec/
[4] https://github.com/facebookresearch/fastText
[5] https://github.com/stanfordnlp/GloVe
[6] https://github.com/Gorov/JointRCM
[7] https://github.com/Bollegala/jointreps
[8] https://github.com/mfaruqui/retrofitting
[9] http://wacky.sslmit.unibo.it

Table 2: Comparison against the works learning word embedding for a corpus

	Semantic [%]	Syntactic [%]	Total[%]
Ours	**73.29**	59.44	**65.85**
CBOW	72.65	59.25	65.33
SG	72.26	55.37	63.04
Enriched CBOW	33.08	**75.39**	56.19
Enriched SG	61.66	64.48	63.20
GloVe	66.35	43.46	53.80

Table 3: Comparison against the previous works learning word embedding using a lexicon

	Semantic [%]	Syntactic [%]	Total[%]
Ours	**73.29 (+0.64)**	59.44 (+0.19)	**65.85** (+0.52)
JointRCM	-	29.9 (-30)	-
JointReps	61.46 (-4.89)	**69.33 (+25.87)**	65.76 **(+11.96)**
RC-Net	34.36 (-37.9)	44.42 (-10.95)	-
Retro (CBOW)	53.88 (-18.77)	61.31 (+2.06)	57.94 (-7.39)
Retro (SG)	50.66 (-21.6)	59.78 (+4.41)	55.64 (-7.4)

of a country. The syntactic part is about syntactic relationships, such as predicting the adverb form for an adjective.

In Table 2, we compare our method with the works using only corpus to train word embeddings. Our method gets the best overall accuracy and the best for the semantic part. For the syntactic part, our method fails to outperform the word2vec enriched with character-level subword information that is reported powerful for the syntactic part.

In Table 3, we compare our method with the previous works using the lexicon to improve the word embeddings. The numbers in the brackets are the differences from the accuracies achieved by the models they base on. For our method and jointRCM, it is CBOW. For jointReps, it is GloVe. For RC-Net, it is SG. For Retro, we report the results retrofitting the word embeddings trained by CBOW and SG, respectively.

We see that while the other works perform worse in the semantic part than the model they base on, ours outperform CBOW and outperform the other works in the semantic part. Benefited from alleviating the bad influence of polysemantic words and utterances, our method successfully improves the word embeddings in representing semantic information using a lexicon while the other works fail to achieve it. Our method also achieves the best overall accuracy. But for syntactic part, the result of our method using enwiki9 is not as good as the reported result of jointReps using ukWaC.

4.4 Effects of the size of the corpus

To see how the size of the corpus affects the performance of our method, we also used a smaller corpus called Text8[10] to learn word embeddings using our methods and then run the word analogical reasoning task using the trained word embeddings. Text8 contains the words in the first 100 million bytes of English Wikipedia. There are 16,718,843 tokens in it and 71,291 different words among them.

[10]http://mattmahoney.net/dc/text8.zip

Table 4: The accuracy of our method and the original CBOW [15] in word analogical reasoning task under different corpus size

		Ours	CBOW	**Difference**
Text8 (17M Tokens)	Semantic[%]	46.35	46.72	-0.37
	Syntactic[%]	42.13	41.90	**+0.23**
	Total[%]	43.88	43.91	-0.03
Enwiki9 (123M Tokens)	Semantic[%]	73.29	72.65	**+0.64**
	Syntactic[%]	59.44	59.25	**+0.19**
	Total[%]	65.72	65.33	**+0.52**

Table 5: Comparison of the learning speed

	Ours	CBOW	Enriched CBOW	JointRCM
Time Cost	3m13s	3m6s	18m	-

We compare the difference of the results by our method and CBOW using the different corpora in Table 4. We see that our method does not achieve obvious improvement over CBOW for text8, but outperforms CBOW for enwiki9. It indicates that our method is weak at small corpus. It is because we use a probabilistic method that requires plenty of samples. By increasing the size of the corpus, our method achieves more improvement.

4.5 The learning speed

We compare the learning speed on our machine of our method training 200-dimension word embeddings on text8 against CBOW and the other related works on the basis of CBOW in Table 5. 20 threads were used to train the word embeddings for every model. Unfortunately, the public implement of jointRCM by the original authors[11] fails to run correctly on our machine, and there is no reported learning speed. We see that there is almost no loss for our method in learning speed in comparison with CBOW while the word2vec enriched with subword information is obviously slower.

5 Conclusion

We figure out a problem that is not paid enough attention to in the previous works using lexicons to improve word embeddings: Because some words and utterances have multiple meanings, a paraphrase of a word in a lexicon may not be a paraphrase actually in a certain context. Then we propose a method to avoid the trap: We treat the lexicon as a fuzzy set, approximately estimate the membership of the paraphrases, and learn word embeddings using both a corpus and a lexicon by replacing the words in the context with the paraphrases randomly subject to their grades of membership.

By comparison with the previous works in the word analogical reasoning task, it has been shown that our method overcomes the weakness of the previous related works in extracting the semantic features, outperforms the previous works and keeps fast.

The results using corpora in different sizes show that the proposed method works better with a larger corpus but less effectively with a small corpus. We are looking for another robust method to control the replacements of the paraphrases that keeps efficient for small corpora.

[11] https://github.com/Gorov/JointRCM

References

[1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155, 2003.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[3] Danushka Bollegala, Alsuhaibani Mohammed, Takanori Maehara, and Ken-Ichi Kawarabayashi. Joint word representation learning using a corpus and a semantic lexicon. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, 2016.

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

[5] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian lda for topic models with word embeddings. 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)*. Association for Computational Linguistics, 2015.

[6] Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*, 2015.

[7] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pp. 758–764, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[8] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 873–882. Association for Computational Linguistics, 2012.

[9] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[10] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[11] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *the 31st International Conference on Machine Learning (ICML 2014)*, volume 14, pp. 1188–1196, 2014.

[12] Shaohua Li, Tat-Seng Chua, Jun Zhu, and Chunyan Miao. Generative topic embedding: a continuous representation of documents. In *the 54th annual meeting of the Association for Computational Linguistics (ACL 2016)*. Association for Computational Linguistics, 2016.

[13] Bill MacCartney. *NATURAL LANGUAGE INFERENCE*. PhD thesis, Stanford University, 2009.

[14] Tomas Mikolov. *Statistical Language Models Based on Neural Networks*. PhD thesis, Brno University of Technology, 2012.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR Workshop*, 2013.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[17] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL HLT*, volume 13, pp. 746–751, 2013.

[18] Andriy Mnih and Geoffrey Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pp. 641–648. ACM, 2007.

[19] Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pp. 1081–1088, 2009.

[20] Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword fifth edition. Technical report, Linguistic Data Consortium, 2011.

[21] Ellie Pavlick, Johan Bos, Malvina Nissim, Charley Beller, Benjamin Van Durme, and Chris Callison-Burch. Adding semantics to data-driven paraphrasing. In *Association for Computational Linguistics*, Beijing, China, July 2015. Association for Computational Linguistics.

[22] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevich, Benjamin Van Durme, and Chris Callison-Burch. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Association for Computational Linguistics*, Beijing, China, July 2015. Association for Computational Linguistics.

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

[24] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1201–1211. Association for Computational Linguistics, 2012.

[25] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, 2014.

[26] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics (ACL 2010)*, pp. 384–394. Association for Computational Linguistics, 2010.

[27] Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 1219–1228. ACM, 2014.

[28] Mo Yu and Mark Dredze. Improving lexical embeddings with semantic knowledge. In *the 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014)*, pp. 545–550. Association for Computational Linguistics, 2014.

[29] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *CoRR*, abs/1409.2329, 2014.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Introduction states the claims.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Sections 4.5 and 5 (Conclusion) some limitations are presented.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No theoretical results are presented.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental section provides a corpus and baselines.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The corpus is open and the code is provided on Github.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All details are given in the experiment section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The result tables include error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Learning speed is discussed in Section 4.5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We are aware of the code of Ethics and read it. All is conform.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss this in the Related Work section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our research does not present such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All is conform.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No new assets are provided.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No research using crowdsourcing.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.