Delta Embedding Learning

Xiao Zhang* **Ji Wu*** **Dejing Dou†**
*Department of Electronic Engineering, Tsinghua University
†Department of Computer and Information Science, University of Oregon
xiphzx@gmail.com
wuji_ee@mail.tsinghua.edu.cn
dou@cs.uoregon.edu

Abstract

Unsupervised word embeddings have become a popular approach of word representation in NLP tasks. However there are limitations to the semantics represented by unsupervised embeddings, and inadequate fine-tuning of embeddings can lead to suboptimal performance. We propose a novel learning technique called *Delta Embedding Learning*, which can be applied to general NLP tasks to improve performance by optimized tuning of the word embeddings. A structured regularization is applied to the embeddings to ensure they are tuned in an incremental way. As a result, the tuned word embeddings become better word representations by absorbing semantic information from supervision without "forgetting." We apply the method to various NLP tasks and see a consistent improvement in performance. Evaluation also confirms the tuned word embeddings have better semantic properties.

1 Introduction

Unsupervised word embeddings have become the basis for word representation in NLP tasks. Models such as skip-gram (Mikolov et al., 2013a) and Glove (Pennington et al., 2014) capture the statistics of a large corpus and have good properties that corresponds to the semantics of words (Mikolov et al., 2013b). However there are certain problems with unsupervised word embeddings, such as the difficulty in modeling some fine-grained word semantics. For example words in the same category but with different polarities are often confused because those words share common statistics in the corpus (Faruqui et al., 2015; Mrkšić et al., 2016).

In supervised NLP tasks, these unsupervised word embeddings are often used in one of two ways: keeping fixed or using as initialization (fine-tuning). The decision is made based on the amount of available training data in order to avoid overfitting. Nonetheless, underfitting with keeping fixed and certain degrees of overfitting with fine-tuning is inevitable. Because this all or none optimization of the word embeddings lacks control over the learning process, the embeddings are not trained to an optimal point, which can result in suboptimal task performance, as we will show later.

In this paper, we propose *delta embedding learning*, a novel method that aims to address the above problems together: using regularization to find the optimal fine-tuning of word embeddings. Better task performance can be reached with properly optimized embeddings. At the same time, the regularized fine-tuning effectively combines semantics from supervised learning and unsupervised learning, which addresses some limitations in unsupervised embeddings and improves the quality of embeddings.

Unlike retrofitting (Yu and Dredze, 2014; Faruqui et al., 2015), which learns directly from lexical resources, our method provides a way to learn word semantics from supervised NLP tasks. Embeddings usually become task-specific and lose its generality when trained along with a model to maximize a task objective. Some approach tried to learn reusable embeddings from NLP tasks include multi-task learning, where one predicts context words and external labels at the same time (Tang et al., 2014), and specially designed gradient descent algorithms for fine-tuning (Yang and Mao, 2015). Our method learns reusable supervised embeddings by fine-tuning an unsupervised embeddings on a supervised task with a simple modification. The method also makes it possible to examine and interpret the learned semantics.

The rest of the paper is organized as follows. Section 2 introduces the *delta embedding learning* method. Section 3 applies the method to NLP tasks, and the learned embeddings are evaluated and analyzed in section 4.

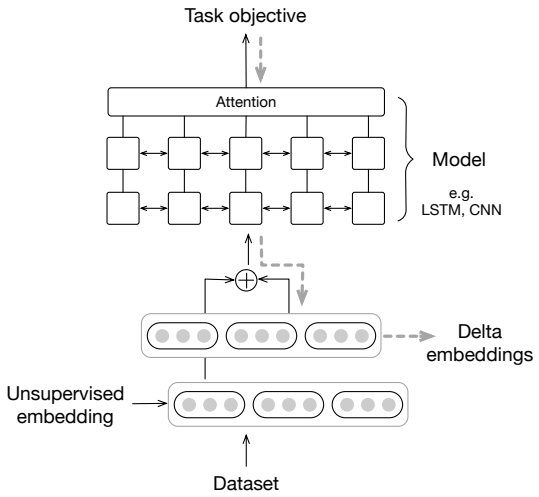

Figure 1: Delta embedding learning in a supervised NLP task. Solid line: forward model computation. Dashed line: learning of delta embeddings through back propagation

2 Methodology

2.1 Delta embedding learning

The aim of the method is to combine the benefits of unsupervised learning and supervised learning to learn better word embeddings. An unsupervised word embeddings like skip-gram, trained on a large corpus (like Wikipedia), gives good-quality word representations. We use such an embedding $\mathbf{w}_{unsup}$ as a starting point and learn a delta embedding $\mathbf{w}_\Delta$ on top of it:

$$\mathbf{w} = \mathbf{w}_{unsup} + \mathbf{w}_\Delta. \quad (1)$$

The unsupervised embedding $\mathbf{w}_{unsup}$ is fixed to preserve good properties of the embedding space and the word semantics learned from large corpus. Delta embedding $\mathbf{w}_\Delta$ is used to capture discriminative word semantics from supervised NLP tasks and is trained together with a model for the supervised task. In order to learn only useful word semantics rather than task-specific peculiarities that results from fitting (or overfitting) a specific task, we impose L_{21} loss, one kind of structured regularization on $\mathbf{w}_\Delta$:

$$loss = loss_{task} + c\sum_{i=1}^{n}(\sum_{j=1}^{d} w_{\Delta ij}^2)^{\frac{1}{2}} \quad (2)$$

The regularization loss is added as an extra term to the loss of the supervised task.

The effect of L_{21} loss on $\mathbf{w}_\Delta$ has a straightforward interpretation: to minimize the total moving distance of word vectors in embedding space while reaching optimal task performance. The L_2 part of the regularization keeps the change of word vectors small, so that it does not lose its original semantics. The L_1 part of the regularization induces sparsity on delta embeddings, that only a small number of words get non-zero delta embeddings, while the majority of words are kept intact. The combined effect is selective fine-tuning with moderation: delta embedding captures only significant word semantics that is contained in the training data of the task while absent in the unsupervised embedding.

2.2 Task formulation

Delta embedding learning is a general method that theoretically can be applied to any tasks or models that use embeddings. Figure 1 is an illustration of how the method is applied. The combined delta embedding and unsupervised embedding is provided to a model as input. The delta embedding is updated with the model while optimizing the loss function in (2). The model is trained to maximize task performance, and the produced delta embedding when combined with the unsupervised embedding becomes an improved word representation in its own right.

3 Experiments on NLP tasks

We conduct experiments on several different NLP tasks to illustrate the effect of delta embedding learning on task performance.

3.1 Experimental setup

Sentiment analysis We performed experiments on two sentiment analysis datasets: rt-polarity (binary) (Pang and Lee, 2005) and Kaggle movie review (KMR, 5 class) (Socher et al., 2013). For rt-polarity, we used a CNN model as in (Kim, 2014). For KMR an LSTM-based model is used.

Reading comprehension We used the Stanford Question Answering Dataset (SQuAD, v1.1) (Rajpurkar et al., 2016) and the Bi-directional Attention Flow (BiDAF) (Seo et al., 2016) model. The original hyperparameters are used, except that character-level embedding is turned off to help clearly illustrate the effect of word embeddings.

Language inference The MultiNLI (Williams et al., 2018) and SNLI (Bowman et al., 2015) datasets are used for evaluation of the natural language inference task. We use the ESIM model,

Regularization coefficient	rt-polarity	KMR	SQuAD		MultiNLI			SNLI
			EM	F1	Genre	M	Mis-M	
0 (finetune)	78.61	68.43	64.29	74.35	69.3	61.2	62.1	57.2
∞ (fixed)	76.66	66.72	67.94	77.33	69.5	62.6	64.0	59.7
10^{-3}	76.17	67.01	68.08	77.56	69.5	63.0	63.6	60.2
10^{-4}	**79.30**	67.97	**68.45**	**78.12**	**71.5**	63.4	**64.3**	**60.6**
10^{-5}	78.71	**68.96**	66.48	76.31	70.9	**63.6**	63.8	59.7

Table 1: Performance of different embedding training methods on various NLP tasks. Numbers represent model accuracy (in percentage) on each task , except for SQuAD

a strong baseline in (Williams et al., 2018). As MultiNLI is a large dataset, we use a subset ("fiction" genre) for training to simulate a moderate data setting, and use development set and SNLI for testing.

Common setup For all the experiments, we used Glove embeddings pre-trained on Wikipedia and Gigaword corpus[1] as they are publicly available and frequently used in NLP literature. Dimensions of word embeddings in all models are set to 100.

3.2 Results

The task performance of models with different embedding learning choices is reported in Table 1. All initialized with unsupervised pre-trained embeddings, comparison is made between fine-tuning, keeping fixed and tuning with delta embeddings. For delta embeddings, there is one hyperparameter c that controls the strength of regularization. We empirically experiment in the range of $[10^{-5}, 10^{-3}]$.

In all the tasks delta embedding learning outperforms conventional methods of using embedding. As embeddings is the only variable, it shows delta embedding learning learns better quality embeddings that results in better task performance.

Roughly two kinds of scenarios exist in these tasks. For easier tasks like sentiment analysis underfitting is obvious when keeping embeddings fixed. Harder tasks like reading comprehension on the other hand clearly suffer from overfitting. In both situations delta embeddings managed to balance between underfitting and overfitting with a more optimal tuning.

For the hyper-parameter choice of regularization coefficient c, we found it fairly insensitive to tasks, with $c = 10^{-4}$ achieving the best performance in most tasks.

[1]https://nlp.stanford.edu/projects/glove/

The results indicate that delta embedding learning does not require the decision to fix the embedding or not in an NLP task, as delta embedding learning always harvests the best from unsupervised embeddings and supervised fine-tuning, regardless of the amount of labeled data.

4 Embedding evaluation

To validate the hypothesis that better performance is the result of better embeddings, we examine the properties of embeddings tuned with delta embedding learning. Word embedding from the BiDAF model is extracted after training on SQuAD, and is compared with the original Glove embedding.

The motivation of investigating embeddings trained on SQuAD is because reading comprehension is a comprehensive language understanding task that involves a rather wide spectrum of word semantics. Training on SQuAD tunes a number of word embeddings which results in non-trivial changes of embedding properties on the whole vocabulary level, which we can validate with embedding evaluation tests. As for simpler tasks like sentiment analysis, we observe that they tune fewer words and the effects are less visible.

4.1 QVEC

QVEC (Tsvetkov et al., 2015) is a comprehensive evaluation of the quality of word embeddings by aligning with linguistic features. We calculated the QVEC score of learned embeddings (Table 2).

Embedding	QVEC score	Relative gain
Glove	0.37536	-
finetune	0.37267	$-2.7 \cdot 10^{-3}$
delta@10^{-3}	0.37536	$3.0 \cdot 10^{-6}$
delta@10^{-4}	**0.37543**	$7.5 \cdot 10^{-5}$
delta@10^{-5}	0.37332	$-2.0 \cdot 10^{-3}$

Table 2: QVEC scores of learned embeddings

Using the original Glove embedding as reference, unconstrained finetune decreases the QVEC

Correlation	Glove	finetune	delta @10^{-4}	Δ
WS-353	0.555	0.545	**0.563**	+
WS-353-SIM	0.657	0.659	**0.667**	+
WS-353-REL	0.495	0.485	**0.506**	+
MC-30	0.776	0.764	**0.783**	+
RG-65	**0.741**	0.736	0.740	-
Rare-Word	0.391	0.377	**0.392**	+
MEN	0.702	**0.703**	**0.703**	+
MTurk-287	0.632	0.625	**0.635**	+
MTurk-771	0.574	**0.577**	0.576	+
YP-130	0.460	**0.475**	0.467	+
SimLex-999	0.292	**0.304**	0.295	+
Verb-143	0.302	0.305	**0.315**	+
SimVerb-3500	0.169	**0.176**	0.171	+

Table 3: Evaluation of embedding by word pair similarity ranking.

score, because the embedding overfits to the task, and some of the semantic information in the original embedding is lost. Delta embedding learning ($c = 10^{-4}$) achieves the best task performance while also slightly increases the QVEC score. The change in score is somewhat marginal, but can be regarded as a sanity check: delta embedding learning does not lower the quality of the original embedding (in other words, it does not suffer from catastrophic forget). Also, as the QVEC score is strongly related to downstream task performance, it also means that delta-tuned embedding is no less general and universal than the original unsupervised embedding.

4.2 Word similarity

Word similarity is a common approach for examining semantics captured by embeddings. We used the tool in (Faruqui and Dyer, 2014) to evaluate on 13 word similarity datasets.

Showed in Table 3, delta embedding trained with $c = 10^{-4}$ has the best performance in over half of the benchmarks. When compared to the original Glove embedding, unconstrained fine-tuned embedding gets better at some datasets while worse at others, indicating that naive fine-tuning learns some semantic information from the task while "forgetting" some others. Delta embedding learning however, achieves better performance than Glove embedding in all but one datasets (negligible decrease on RG-65, see the last column of Table 3). This shows that delta embedding learning effectively learns new semantics from a supervised task and adds it to the original embedding in a non-destructive way. The quality of embedding is improved.

Sentiment Analysis	neither still unexpected nor bore lacking worst suffers usual moving works interesting tv fun smart
Reading Comprehension	why another what along called whose call which also this if not occupation whom but he because into
Language Inference	not the even I nothing because that you it as anything only was if want forget well be so from does in certain could

Table 4: Words with the largest norm of delta embedding in different tasks

4.3 Interpreting word semantics learning

The formulation of delta embeddings makes it possible to help analyze word semantics learned in a supervised task, regardless of the model used.

To answer the question "What is learned in the task?", the norm of delta embeddings can be used to identify which word has a significant newly learned component. In Table 4, for instance, words with a sentiment like "bore" and "fun" are mostly learned in sentiment analysis tasks. In reading comprehension, question words like "what" and "why" are the first to be learned , after that are words helping to locate possible answers like "called," "another," and "also."

	Nearest neighbors of word "*not*"[2]
Before training	(+) good always clearly definitely well able (-) nothing yet none
After training	(+) sure (-) nothing yet none bad lack unable nobody less impossible unfortunately Not rarely

Table 5: The position shift of word "not" in embedding space

The semantics learned in a word can be represented by its shift of position in the embedding space (which is the delta embedding). We found the semantics learned are often discriminative features. Use the word "not" as an example, after training it clearly gains a component representing negativity, and differentiates positive and negative words much better (Table 5). These discriminative semantics are sometimes absent or only weakly present in co-occurrence statistics, but play a crucial role in the understanding of text in NLP tasks.

[2]only showing words with a polarity

5 Conclusion

We proposed delta embedding learning, a supervised embedding learning method that not only improves performance in NLP tasks, but also learns better universal word embeddings by letting the embedding "grow" under supervision.

Because delta embedding learning is an incremental process, it is possible to learn from a sequence of tasks, essentially "continuous learning" (Parisi et al., 2018) of word semantics. It is an interesting future work and will make learning word embeddings more like human learning a language.

Acknowledgments

This research is partially supported by the National Key Research and Development Program of China (No.2018YFC0116800) and the NSF grant CNS-1747798 to the IUCRC Center for Big Learning.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615. Association for Computational Linguistics.

Manaal Faruqui and Chris Dyer. 2014. Community evaluation and exchange of word vectors at wordvectors.org. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 19–24. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations 2013*.

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

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124. Association for Computational Linguistics.

German Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2018. Continual lifelong learning with neural networks: A review. *Neural Networks*.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations 2017*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642. Association for Computational Linguistics.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565. Association for Computational Linguistics.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Xuefeng Yang and Kezhi Mao. 2015. Supervised fine tuning for word embedding with integrated knowledge. *arXiv preprint arXiv:1505.07931*.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 545–550. Association for Computational Linguistics.