Evaluating Word Embeddings Using a Representative Suite of Practical Tasks

Neha Nayak
Stanford University
Stanford, CA 94305

Gabor Angeli
Stanford University
Stanford, CA 94305

Christopher D. Manning
Stanford University
Stanford, CA 94305

{nayakne, angeli, manning}@cs.stanford.edu

Abstract

Word embeddings are increasingly used in natural language understanding tasks requiring sophisticated semantic information. However, the quality of new embedding methods is usually evaluated based on simple word similarity benchmarks. We propose evaluating word embeddings *in vivo* by evaluating them on a suite of popular downstream tasks. To ensure the ease of use of the evaluation, we take care to find a good point in the tradeoff space between (1) creating a thorough evaluation – i.e., we evaluate on a diverse set of tasks; and (2) ensuring an easy and fast evaluation – by using simple models with few tuned hyperparameters. This allows us to release this evaluation as a standardized script and online evaluation, available at http://veceval.com/.

1 Introduction

Many modern NLP systems, especially those employing deep learning techniques, benefit from word representations in the form of low-dimensional word embeddings. This has led to a burgeoning body of work focusing on improving these representations. These word representations are used either as features for a conventional statistical classifier, or in a deep learning setup, where they are tuned to a particular task through back-propagation.

However, the quality of these unsupervised embeddings is often asserted on the basis of restricted lexical semantics tasks, such as scoring word similarity or linear relationships for analogies. These intrinsic evaluations are carried out with little attention paid to how performance correlates with downstream tasks. We propose instead evaluating word embeddings using a standardized suite of characteristic downstream tasks.

This has two advantages, which constitute the main contributions of the paper. First, an improvement in performance on these representative tasks is more likely to generalize to real-world applications of the embedding, as compared to improvements in performance on current word similarity benchmarks. Therefore, this evaluation offers a better metric to hill-climb on than current lexical semantics tasks.

Second, this evaluation allows for higher fidelity qualitative assessment on the strengths and weaknesses of an embedding method. For instance, certain embeddings may excel at syntactic tasks, or on sequence modeling tasks, whereas others may capture the semantics of a word better, or work better for classification tasks. We believe this evaluation can facilitate consolidating and formalizing such insights, currently latent in the collective consciousness of the NLP community.

2 Related work

Existing work on creating evaluations for word embeddings has focused on lexical semantics tasks. An example of such tasks is WordSim-353 (Finkelstein et al., 2001), in which a series of word pairs are assigned similarity judgments by human annotators, and these are compared to the similarity scores obtained from word embeddings.

A thorough such lexical semantics evaluation was created by Faruqui and Dyer (2014)[1]. This website allows a user to upload a set of embeddings, and evaluates these embeddings on a series of word similarity benchmarks. We follow the model presented in Faruqui and Dyer (2014), but extend to a series of more realistic downstream tasks.

[1] http://www.wordvectors.org

Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP, pages 19–23,
Berlin, Germany, August 12, 2016.

Schnabel et al. (2015) carried out both a thorough intrinsic evaluation of word vectors, and a limited extrinsic evaluation showing that an embedding's intrinsic performance did not necessarily correlate with its real-world performance. This finding is a key motivation for this work – we aim to create a metric which does correlate with downstream performance.

3 Motivation

While extensive research has gone into the development of meaningful intrinsic evaluation methods, extrinsic evaluation remains the de-facto proving ground for novel word embedding methods (Pennington et al., 2014; Dhillon et al., 2012). We aim to create an evaluation methodology which is representative of real-world performance, but nonetheless fast and easy to evaluate against. These two criteria are somewhat at odds with each other, which necessitates finding a good point in a number of key tradeoffs:

Choice of Tasks Optimally, new embeddings would be evaluated on as large a number of tasks as possible. However, such an evaluation would become prohibitively slow and impractical. Therefore, we limit our evaluation to 6 tasks of moderate size, allowing models to be trained and evaluated quickly while nonetheless covering a range of NLP phenomena.

Choice of Models Optimally, new embeddings would be evaluated as components of a range of different models. However, the addition of more models – and in particular more sophisticated models – slows down the evaluation. Therefore, we opt to use uncontroversial off-the-shelf neural models for each task. Although none of the models achieve state-of-the-art results, we only care about *relative* scores between embeddings. Simple models are equally, if not more suitable for this criterion.

Choice of hyperparameters The performance of neural models often vary greatly depending on the choice of hyperparameters. To be fair to everyone, we must either cross-validate for each embedding, or aggressively minimize the number of hyperparameters. For the sake of efficiency, we opt for the latter.

Reproducibility To ensure reproducibility, we release our evaluation script, and host a public website where users can upload their embeddings to be evaluated.

4 Tasks

The following are a selection of tasks to be included in the benchmark suite. These were chosen to be a representative – though certainly not exhaustive – sampling of relevant downstream tasks.

Two tasks are included to test syntactic properties of the word embeddings – part-of-speech tagging and chunking. Part-of-speech tagging is carried out on the WSJ dataset described in Toutanova et al. (2003). In order to simplify the task and avoid hand-coded features, we evaluate against the universal part-of-speech tags proposed in Petrov et al. (2012). For chunking, we use the dataset from the CoNLL 2000 shared task (Tjong Kim Sang and Buchholz, 2000), derived from the Wall Street Journal.

Four tasks test the semantic properties of the word embeddings. At the word level, we include named entity recognition. We evaluate on a 4-class Named Entity Recognition task: PERSON, LOCATION, ORGANIZATION, and MISC, using the CoNLL 2003 dataset (Tjong Kim Sang and De Meulder, 2003), and an IOB tagging scheme. At the sentence level, we include two tasks – sentiment classification and question classification. We implement binary sentiment classification using the Stanford Sentiment Treebank dataset, and the coarse-grained question classification task described in Li and Roth (2006).

Finally, above the word level, we test the ability of word embeddings to propagate the lexical relation information they contain into representations of larger units of text. This involves the task of phrase-level natural language inference, derived from a dataset presented in Ganitkevitch et al. (2013).

These tasks were selected so as to cover both syntactic and semantic capabilities, but also as they are fast to train, fulfilling another of the characteristics put forward in Section 3.

5 Models

Our goal is to select the simplest possible models for each task, while maintaining respectable performance. We therefore train straightforward models using standard neural net layers.

For tasks with word-level labeling, we use a window-based model akin to that in Collobert et

al. (2011). Features for each token in a sequence are constructed by concatenating the word embeddings of a window of words centered on the token. This is passed through a two-layer neural network, followed by a softmax classifier.

For tasks with sentence-level labeling, sentence representations are constructed using a basic LSTM. Classification is then carried out by passing through a one-layer neural network, followed by a softmax classifier.

Finally, the NLI task requires representations for both the premise and hypothesis sequences. Embeddings for each sequence are constructed as described in the sentence embedding tasks, using two separate LSTMs. These embeddings are concatenated, and similarly passed through a one-layer neural network and softmax classifier. Our implementations of these simple models are able to train with a new set of embeddings and evaluate the resulting model in a few hours.

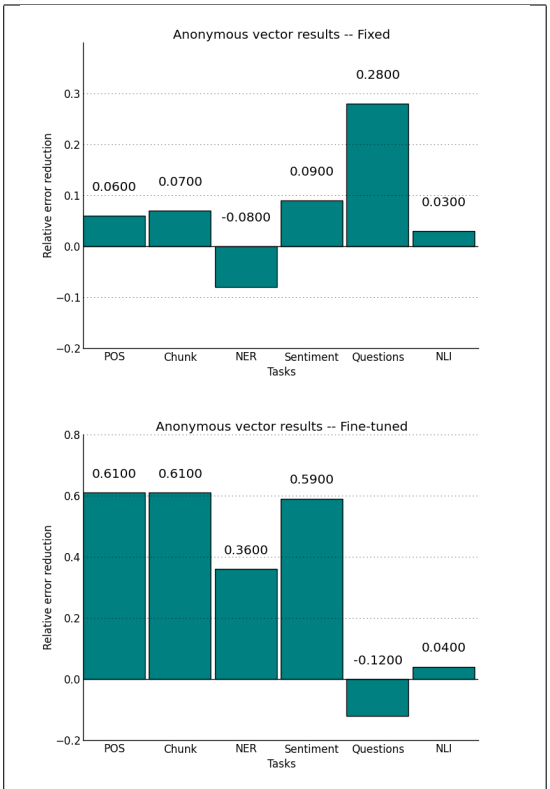
Although these simplistic models do not achieve state-of-the-art performance on any of the tasks, they are faster and in many ways more robust to variations in training methodology than more sophisticated models, maintaining a reduced hyperparameter set. Furthermore, a valid comparison between word vectors requires only that the model is *fair* to each representation, not necessarily that the models achieve state-of-the-art performance, fulfilling our requirements from Section 3.

This evaluation aims solely to test the properties of word embeddings, and not phrase or sentence embeddings. For the tasks that demand phrase and sentence representations, we elect to construct these from the word embeddings using an LSTM, rather than to extend the evaluation to other types of embeddings.

6 Evaluation metrics

Our goal is to distill performance on extrinsic tasks into a short but comprehensive "report" that indicates the strengths and weaknesses of a particular set of embeddings on a variety of downstream tasks. For each set of embeddings tested, we report results based on the metric most appropriate for the task – F_1 score for NER, and accuracy for the rest of the tasks.

We use SVD as a baseline embedding method. Using the hyperwords software of Levy et al. (2015), we apply SVD to a PMI-transformed co-occurrence matrix derived from the same pre-



Raw results on downstream tasks

	POS (Acc.)	Chunk (Acc.)	NER (F1)	Sentiment (Acc.)	Questions (Acc.)	NLI (Acc.)
Fixed	93.30	91.20	96.30	87.40	82.10	48.40
(Baseline)	82.80	77.60	94.20	69.20	84.00	46.40
Fine-tuned	97.10	93.30	95.80	78.70	88.80	45.60
(Baseline)	96.90	92.80	96.10	76.70	84.40	43.90

Raw results on word similarity tasks

	WordSim	Analogy (Sem.)	Analogy (Syn.)
Anonymous vectors	0.640	52.0	63.0
Baseline	0.560	38.0	48.0

Figure 1: An example of the type of result report created by our evaluation. The first chart shows the relative error reduction of the embedding method compared to the SVD baseline, *disallowing* backpropagation into the vectors. This measures the extent to which the original vectors capture linguistic phenomena. Values above 0 perform better than SVD on the task; the magnitude of the improvement are on comparable scales between tasks. The second chart is identical to the first chart, but allowing backpropagation into the vectors. This measures how good the vectors are as an initialization for neural network methods. The first table shows the raw accuracy numbers for each task. The second table shows the vectors' result on the WordSim and Analogy tasks.

scribed corpus, resulting in a set of SVD vectors. We present a baseline for each task, which is the F_1 score or accuracy attained by the SVD vectors.

Due to the diversity of the tasks, it is difficult to compare the raw values or differences over the baseline. These measures, especially when aggregated, tend to implicitly reward large improvements over low-baseline tasks more than small improvements over high-baseline tasks. To illustrate, whereas a 1% improvement on POS tagging should be considered significant, the same 1% improvement on a task with a 80% baseline is less impressive. As such, the primary metric we report is not accuracy or F_1, but rather the *relative error reduction* as compared to the SVD baseline. This allows us to calculate a meaningful aggregate, averaging relative error reduction over tasks. For backwards compatibility with prior work, we also report correlations on WordSim-353, as well as precision at 1 for the analogy task presented in Mikolov et al. (2013).

The figure shows an example report generated by our evaluation, using arbitrary but realistic values. It can be seen that the relative error reduction depicted in the charts enables a clearer representation of the relative performance on different tasks, as compared to the raw values provided in the table.

7 Experimental methodology

7.1 Training hyperparameters

Following Schnabel et al. (2015), we prescribe the use of a fixed snapshot of Wikipedia (dated 2008-03-01) for training the embeddings to be evaluated. This corpus was selected to be as close in time as possible to the corpus Collobert et al. (2011)'s embeddings were trained on. It was preprocessed by applying the Stanford tokenizer (Manning et al., 2014), and replacing all digits with zeros.

7.2 Avoiding bias

Since this method of evaluation involves training a number of neural network models, there is a significant danger of overfitting to the embeddings used to find the hyperparameters. We attempt to mitigate this in two ways.

First, we use simple models with standard neural net layers to limit the number of hyperparameters tuned. We tune only the optimizer type, the l2 coefficient for regularization, and the learning rate. We set any additional hyperparameters to fast yet reasonable defaults, which also facilitate short training times. For example, in an LSTM layer, we use a hidden layer size equal to the input vector size. Second, rather than optimizing for each individual task, we select only two hyperparameter settings – one for the sequence labelling tasks (POS tagging, chunking and NER), and a separate setting for the other tasks. This is necessitated by the difference in model structure.

7.3 Fine-tuning

Most deep learning-based models backpropagate into the word embeddings used so as to fine tune them to the task at hand. This is a realistic setting in which to examine the performance of word embeddings, in their capacity as an initialization for the various tasks. In contrast, disallowing backpropagation into the embeddings allows us to determine the amount of syntactic or semantic information inherently present in the embeddings. As such, we propose reporting accuracies attained in both these settings.

8 Practical details

Evaluation takes place on the web site `http://www.veceval.com`. It is assumed that the user will train word embeddings to be evaluated, using the corpus provided on the website. The sentence and phrase embeddings used in the evaluation are produced by composing these given word embeddings. The user is required to prepare a gzipped text file, containing the word embeddings to be evaluated, in a simple format specified on the website. When the file is uploaded to the website, evaluation will begin. Once the evaluation is complete, a link to a report of the embeddings' performance appears on the homepage.

It is expected that the evaluation will take a few hours. For example, the best performing hyperparameters on the baseline embeddings result in a running time of 4 hours and 24 minutes.

9 Conclusion

We have presented a proposal for a fair and replicable evaluation for word embeddings. We plan to make this evaluation available as a script, allowing it to be run on new embeddings. It is our hope that this benchmark will enable extrinsic evaluations to be compared in a more interpretable way.

References

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

Paramveer Dhillon, Jordan Rodu, Dean Foster, and Lyle Ungar. 2012. Two step cca: A new spectral method for estimating vector models of words. *arXiv preprint arXiv:1206.6403*.

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

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *HLT-NAACL*.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Xin Li and Dan Roth. 2006. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(03):229–249.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of LREC*, May.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Erik F Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *CoNLL*.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *HLT-NAACL*.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180.