Evaluation methods for unsupervised word embeddings

Tobias Schnabel
Cornell University
Ithaca, NY, 14853
tbs49@cornell.edu

Igor Labutov
Cornell University
Ithaca, NY, 14853
iil4@cornell.edu

David Mimno, Thorsten Joachims
Cornell University
mimno@cornell.edu,
tj@cs.cornell.edu

Abstract

We present a comprehensive study of evaluation methods for unsupervised embedding techniques that obtain meaningful representations of words from text. Different evaluations result in different orderings of embedding methods, calling into question the common assumption that there is one single optimal vector representation. We present new evaluation techniques that directly compare embeddings with respect to specific queries. These methods reduce bias, provide greater insight, and allow us to solicit data-driven relevance judgments rapidly and accurately through crowdsourcing.

1 Introduction

Neural word embeddings represent meaning via geometry. A good embedding provides vector representations of words such that the relationship between two vectors mirrors the linguistic relationship between the two words. Despite the growing interest in vector representations of semantic information, there has been relatively little work on direct evaluations of these models. In this work, we explore several approaches to measuring the quality of neural word embeddings. In particular, we perform a comprehensive analysis of evaluation methods and introduce novel methods that can be implemented through crowdsourcing, providing better insights into the relative strengths of different embeddings.

Existing schemes fall into two major categories: extrinsic and intrinsic evaluation. In extrinsic evaluation, we use word embeddings as input features to a downstream task and measure changes in performance metrics specific to that task. Examples include part-of-speech tagging and named-entity recognition (Pennington et al., 2014). Extrinsic evaluation only provides one way to specify the goodness of an embedding, and it is not clear how it connects to other measures.

Intrinsic evaluations directly test for syntactic or semantic relationships between words (Mikolov et al., 2013a; Baroni et al., 2014). These tasks typically involve a pre-selected set of query terms and semantically related target words, which we refer to as a *query inventory*. Methods are evaluated by compiling an aggregate score for each method such as a correlation coefficient, which then serves as an absolute measure of quality. Query inventories have so far been collected opportunistically from prior work in psycholinguistics, information retrieval (Finkelstein et al., 2002), and image analysis (Bruni et al., 2014). Because these inventories were not constructed for word embedding evaluation, they are often idiosyncratic, dominated by specific types of queries, and poorly calibrated to corpus statistics.

To remedy these problems, this paper makes the following contributions. First, this is the first paper to conduct a comprehensive study covering a wide range of evaluation criteria and popular embedding techniques. In particular, we study how outcomes from three different evaluation criteria are connected: word relatedness, coherence, downstream performance. We show that using different criteria results in different relative orderings of embeddings. These results indicate that embedding methods should be compared in the context of a specific task, e.g., linguistic insight or good downstream performance.

Second, we study the connections between direct evaluation with real users and pre-collected offline data. We propose a new approach to evaluation that focuses on direct comparison of embeddings with respect to individual queries rather than overall summary scores. Because we phrase all tasks as choice problems rather than ordinal relevance tasks, we can ease the burden of the an-

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 298–307,
Lisbon, Portugal, 17-21 September 2015.

notators. We show that these evaluations can be gathered efficiently from crowdsourcing. Our results also indicate that there is in fact strong correlation between the results of automated similarity evaluation and direct human evaluation. This result justifies the use of offline data, at least for the similarity task.

Third, we propose a model- and data-driven approach to constructing query inventories. Rather than picking words in an *ad hoc* fashion, we select query words to be diverse with respect to their frequency, parts-of-speech and abstractness. To facilitate systematic evaluation and comparison of new embedding models, we release a new frequency-calibrated query inventory along with all user judgments at `http://www.cs.cornell.edu/~schnabts/eval/`.

Finally, we observe that word embeddings encode a surprising degree of information about word frequency. We found this was true even in models that explicitly reserve parameters to compensate for frequency effects. This finding may explain some of the variability across embeddings and across evaluation methods. It also casts doubt on the common practice of using the vanilla cosine similarity as a similarity measure in the embedding space.

It is important to note that this work is a survey of *evaluation* methods not a survey of *embedding* methods. The specific example embeddings presented here were chosen as representative samples only, and may not be optimal.

2 Word embeddings

We refer to a word embedding as a mapping $V \rightarrow \mathbb{R}^D : w \mapsto \vec{w}$ that maps a word w from a vocabulary V to a real-valued vector $\vec{w}$ in an embedding space of dimensionality D.

Following previous work (Collobert et al., 2011; Mikolov et al., 2013a) we use the commonly employed cosine similarity, defined as $\text{similarity}(w_1, w_2) = \frac{\vec{w_1} \cdot \vec{w_2}}{\|\vec{w_1}\| \|\vec{w_2}\|}$, for all similarity computations in the embedding space. The list of nearest neighbors of a word w are all words $v \in V \setminus \{w\}$, sorted in descending order by $\text{similarity}(w, v)$. We will denote w as the *query word* in the remainder of this paper.

All experiments in this paper are carried out on six popular unsupervised embedding methods. These embeddings form a representative but incomplete subset; and since we are studying evaluation methods and not embeddings themselves, no attempt has been made to optimize these embeddings. The first two embedding models, the CBOW model of word2vec (Mikolov et al., 2013a) and C&W embeddings (Collobert et al., 2011) both are motivated by a probabilistic prediction approach. Given a number of context words around a target word w, these models formulate the embedding task as that of finding a representation that is good at predicting w from the context representations.

The second group of models, Hellinger PCA (Lebret and Collobert, 2014), GloVe (Pennington et al., 2014), TSCCA (Dhillon et al., 2012) and Sparse Random Projections (Li et al., 2006) follow a reconstruction approach: word embeddings should be able to capture as much relevant information from the original co-occurrence matrix as possible.

Training corpus. We tried to make the comparison as fair as possible. As the C&W embeddings were only available pretrained on a November 2007 snapshot of Wikipedia, we chose the closest available Wikipedia dump (2008-03-01) for training the other models. We tokenized the data using the Stanford tokenizer (Manning et al., 2014). Like Collobert et al. (2011), we lower-cased all words and replaced digits with zeros.

Details. All models embedded words into a 50-dimensional space ($D = 50$). As implemented, each method uses a different vocabulary, so we computed the intersection of the six vocabularies and used the resulting set of 103,647 words for all nearest-neighbor experiments.

3 Relatedness

We begin with intrinsic evaluation of relatedness using both pre-collected human evaluations and a novel online user study. Section 3.1 introduces the list of datasets that is commonly used as a benchmark for embedding methods. There, embeddings are evaluated individually and only their final scores are compared, hence we refer to this scenario as *absolute intrinsic* evaluation. We present a new scenario, *comparative intrinsic* evaluation, in which we ask people directly for their preferences among different embeddings. We demonstrate that we can achieve the same results as offline, absolute metrics using online, comparative metrics.

3.1 Absolute intrinsic evaluation

For the absolute intrinsic evaluation, we used the same datasets and tasks as Baroni et al. (2014). While we present results on all tasks for completeness, we will mainly focus on relatedness in this section. There are four broad categories:

- **Relatedness**: These datasets contain relatedness scores for pairs of words; the cosine similarity of the embeddings for two words should have high correlation (Spearman or Pearson) with human relatedness scores.
- **Analogy**: This task was popularized by Mikolov et al. (2013a). The goal is to find a term x for a given term y so that $x : y$ best resembles a sample relationship $a : b$.
- **Categorization**: Here, the goal is to recover a clustering of words into different categories. To do this, the corresponding word vectors of all words in a dataset are clustered and the purity of the returned clusters is computed with respect to the labeled dataset.
- **Selectional preference**: The goal is to determine how typical a noun is for a verb either as a subject or as an object (e.g., people eat, but we rarely eat people). We follow the procedure that is outlined in Baroni et al. (2014).

Several important design questions come up when designing reusable datasets for evaluating relatedness. While we focus mainly on challenges that arise in the relatedness evaluation task, many of the questions discussed also apply to other scenarios.

Query inventory. How we pick the word pairs to evaluate affects the results of the evaluation. The commonly-used WordSim-353 dataset (Finkelstein et al., 2002), for example, only tries to have word pairs with a diverse set of similarity scores. The more recent MEN dataset (Bruni et al., 2014) follows a similar strategy, but restricts queries to words that occur as annotations in an image dataset. However, there are more important criteria that should be considered in order to create a diverse dataset: (i) the frequency of the words in the English language (ii) the parts of speech of the words and (iii) abstractness vs. concreteness of the terms. Not only is frequency important because we want to test the quality of embeddings on rare words, but also because it is related with distance in the embedding space as we show later and should be explicitly considered.

Metric aggregation. The main conceptual shortcoming of using correlation-based metrics is that they aggregate scores of different pairs — even though these scores can vary greatly in the embedding space. We can view the relatedness task as the task of evaluating a set of rankings, similar to ranking evaluation in Information Retrieval. More specifically, we have one query for each unique query word w and rank all remaining words v in the vocabulary accordingly. The problem now is that we usually cannot directly compare scores from different rankings (Aslam and Montague, 2001) as their scores are not guaranteed to have the same ranges. An even worse case is the following scenario. Assume we use rank correlation as our metric. As a consequence, we need our gold ranking to define an order on all the word pairs. However, this also means that we somehow need to order completely unrelated word pairs; for example, we have to decide whether (dog, cat) is more similar than (banana, apple).

3.2 Absolute results

Table 1 presents the results on 14 different datasets for the six embedding models. We excluded examples from datasets that contained words not in our vocabulary. For the relatedness and selective preference tasks, the numbers in the table indicate the correlation coefficient of human scores and the cosine similarity times 100. The numbers for the categorization tasks reflect the purities of the resulting clusters. For the analogy task, we report accuracy.

CBOW outperforms other embeddings on 10 of 14 datasets. CBOW especially excels at the relatedness and analogy tasks, but fails to surpass other models on the selective preferences tasks. Random projection performs worst in 13 out of the 14 tasks, being followed by Hellinger PCA. C&W and TSCCA are similar on average, but differ across datasets. Moreover, although TSCCA and GloVe perform similarly on most tasks, TSCCA suffers disproportionally on the analogy tasks.

3.3 Comparative intrinsic evaluation

In comparative evaluation, users give direct feedback on the embeddings themselves, so we do not have to define a metric that compares scored word pairs. Rather than defining both query and target words, we need only choose query words since the

	relatedness						categorization			sel. prefs		analogy			
	rg	ws	wss	wsr	men	toefl	ap	esslli	batt.	up	mcrae	an	ansyn	ansem	average
CBOW	**74.0**	**64.0**	**71.5**	**56.5**	**70.7**	66.7	**65.9**	**70.5**	**85.2**	24.1	13.9	**52.2**	**47.8**	**57.6**	**58.6**
GloVe	63.7	54.8	65.8	49.6	64.6	**69.4**	64.1	65.9	77.8	27.0	**18.4**	42.2	44.2	39.7	53.4
TSCCA	57.8	54.4	64.7	43.3	56.7	58.3	57.5	**70.5**	64.2	**31.0**	14.4	15.5	19.0	11.1	44.2
C&W	48.1	49.8	60.7	40.1	57.5	66.7	60.6	61.4	80.2	28.3	16.0	10.9	12.2	9.3	43.0
H-PCA	19.8	32.9	43.6	15.1	21.3	54.2	34.1	50.0	42.0	-2.5	3.2	3.0	2.4	3.7	23.1
Rand. Proj.	17.1	19.5	24.9	16.1	11.3	51.4	21.9	38.6	29.6	-8.5	1.2	1.0	0.3	1.9	16.2

Table 1: Results on absolute intrinsic evaluation. The best result for each dataset is highlighted in bold. The second row contains the names of the corresponding datasets.

embeddings themselves will be used to define the comparable target words.

Query inventory. We compiled a diverse inventory of 100 query words that balance frequency, part of speech (POS), and concreteness. First, we selected 10 out of 45 broad categories from WordNet (Miller, 1995). We then chose an equal number of categories that mostly contained abstract concepts and categories that referred to concrete concepts. Among those categories, we had one for adjectives and adverbs each, and four for nouns and verbs each. From each category, we drew ten random words with the restriction that there be exactly three rare words (i.e., occurring fewer than 2500 times in the training corpus) among the ten.

Details. Our experiments were performed with users from Amazon Mechanical Turk (MTurk) that were native speakers of English with sufficient experience and positive feedback on the Amazon Mechanical Turk framework.

For each of the 100 query words in the dataset, the nearest neighbors at ranks $k \in \{1, 5, 50\}$ for the six embeddings were retrieved. For each query word and k, we presented the six words along with the query word to the users. Each Turker was requested to evaluate between 25 and 50 *items* per task, where an item corresponds to the query word and the set of 6 retrieved neighbor words from each of the 6 embeddings. The payment was between \$0.01 and \$0.02 per item. The users were then asked to pick the word that is most similar according to their perception (the instructions were almost identical to the WordSim-353 dataset instructions). Duplicate words were consolidated, and a click was counted for all embeddings that returned that word. An option "I don't know the meaning of one (or several) of the words" was also provided as an alternative. Table 2 shows an example instance that was given to the Turkers.

Query:	**skillfully**			
	(a)	swiftly	(b)	expertly
	(c)	cleverly	(d)	pointedly

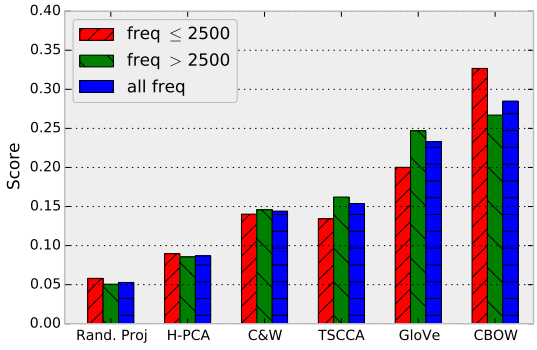
Table 2: Example instance of comparative intrinsic evaluation task. The presented options in this example are nearest neighbors to the query word according to (a) C&W, (b) CBOW, GloVe, TSCCA (c) Rand. Proj. and (d) H-PCA.

The combination of 100 query words and 3 ranks yielded 300 items on which we solicited judgements by a median of 7 Turkers (min=5, max=14). We compare embeddings by average win ratio, where the win ratio was how many times raters chose embedding e divided by the number of total ratings for item i.

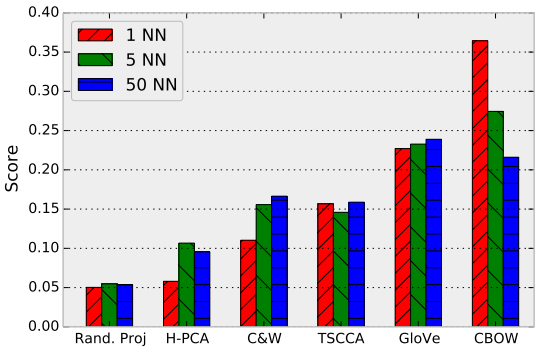
3.4 Comparative results

Overall comparative results replicate previous results. Figure 1(a) shows normalized win ratio scores for each embedding across 3 conditions corresponding to the frequency of the query word in the training corpus. The scores were normalized to sum to one in each condition to emphasize relative differences. CBOW in general performed the best and random projection the worst (p-value < 0.05 for all pairs except H-PCA and C&W in comparing un-normalized score differences for the ALL-FREQ condition with a randomized permutation test). The novel comparative evaluations correspond both in rank and in relative margins to those shown in Table 1.

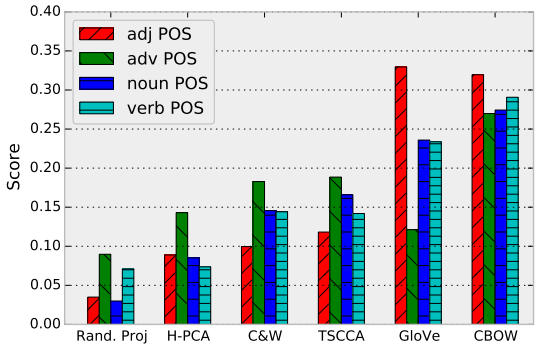
Unlike previous results, we can now show differences beyond the nearest neighbors. Figure 1(b) presents the same results, but this time



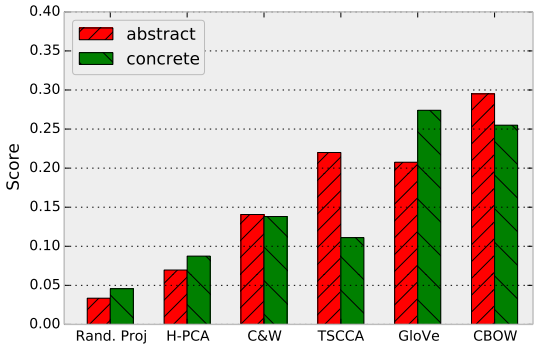
(a) Normalized scores by global word frequency.



(b) Normalized scores by nearest neighbor rank k.



(c) Normalized scores by part of speech.



(d) Normalized scores by category.

Figure 1: Direct comparison task

broken up by the rank k of the neighbors that were compared. CBOW has its strengths especially at rank $k = 1$. For neighbors that appear after that, CBOW does not necessarily produce better embeddings. In fact, it even does worse for $k = 50$ than GloVe. It is important to note, however, that we cannot make absolute statements about how performance behaves across different values of k since each assessment is always relative to the quality of all other embeddings.

We balanced our query inventory also with respect to parts of speech and abstractness vs. concreteness. Figure 1(c) shows the relative performances of all embeddings for the four POS classes (adjectives, adverbs, nouns and verbs). While most embeddings show relatively homogeneous behaviour across the four classes, GloVe suffers disproportionally on adverbs. Moving on to Figure 1(d), we can see a similar behavior for TSCCA: Its performance is much lower on concrete words than on abstract ones. This difference may be important, as recent related work finds that simply differentiating between general and specific terms explains much of the observed variation between embedding methods in hierarchical classification tasks (Levy et al., 2015b). We take the two observations above as evidence that a more fine-grained analysis is necessary in discerning different embedding methods.

As a by-product, we observed that there was no embedding method that consistently performed best on all of the four different absolute evaluation tasks. However, we would like to reiterate that our goal is not to identify one best method, but rather point out that different evaluations (e.g., changing the rank k of the nearest neighbors in the comparison task) result in different outcomes.

4 Coherence

In the relatedness task we measure whether a pair of semantically similar words are near each other in the embedding space. In this novel coherence task we assess whether *groups* of words in a small neighborhood in the embedding space are mutually related. Previous work has used this property for qualitative evaluation using visualizations of 2D projections (Turian et al., 2010), but we are not aware of any work using local neighborhoods for

quantitative evaluation. Good embeddings should have coherent neighborhoods for each word, so inserting a word not belonging to this neighborhood should be easy to spot. Similar to Chang et al. (2009), we presented Turkers with four words, three of which are close neighbors and one of which is an "intruder." For each of the 100 words in our query set of Section 3.3, we retrieved the two nearest neighbors. These words along with the query word defined the set of (supposedly) good options. Table 3 shows an example instance that was given to the Turkers.

(a)	finally	(b)	eventually
(c)	immediately	(d)	put

Table 3: Example instance of intrusion task. The query word is option (a), intruder is (d).

To normalize for frequency-based effects, we computed the average frequency avg of the three words in this set and chose the intruder word to be the first word that had a frequency of $avg \pm 500$ starting at rank 100 of the list of nearest neighbors.

Results. In total, we solicited judgments on 600 items (100 query words for each of the 6 embeddings) from a median of 7 Turkers (min=4, max=11) per item, where each Turker evaluated between 25 and 50 items per task. Figure 2 shows the results of the intrusion experiment. The evaluation measure is micro-averaged precision for an embedding across 100 query words, where per-item precision is defined as the number of raters that discovered the intruder divided the total number of raters of item i. Random guessing would achieve an average precision of 0.25.

All embeddings perform better than guessing, indicating that there is at least some coherent structure captured in all of them. However, the best performing embeddings at this task are TSCCA, CBOW and GloVe (the precision mean differences were not significant under a random permutation test), while TSCCA attains greater precision ($p < 0.05$) in relation to C&W, H-PCA and random projection embeddings. These results are in contrast to the direct comparison study, where the performance of TSCCA was found to be significantly worse than that of CBOW. However, the order of the last three embeddings remains unchanged, implying that performance on the intrusion task and performance on the direct comparison task are correlated. CBOW and C&W seem to do equally well on rare and frequent words, whereas the other models' performance suffers on rare words.

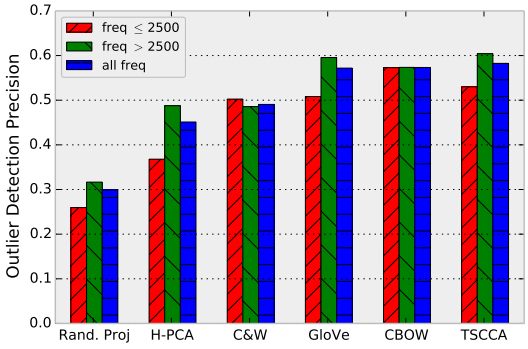


Figure 2: Intrusion task: average precision by global word frequency.

Discussion. Evaluation of set-based properties of embeddings may produce different results from item-based evaluation: rankings we got from the intrusion task did not match the rankings we obtained from the relatedness task. Pairwise similarities seem to be only part of the information that is encoded in word embeddings, so looking at more global measures is necessary for a better understanding of differences between embeddings.

We choose intruder words based on similar but lower-ranked words, so an embedding could score well on this task by doing an unusually bad job at returning less-closely related words. However, the results in Figure 1(b) suggest that there is little differences at higher ranks (rank 50) between embeddings.

5 Extrinsic Tasks

Extrinsic evaluations measure the contribution of a word embedding model to a specific task. There is an implicit assumption in the use of such evaluations that there is a consistent, global ranking of word embedding quality, and that higher quality embeddings will necessarily improve results on any downstream task. We find that this assumption does not hold: different tasks favor different embeddings. Although these evaluations are useful in characterizing the relative strengths of different models, we do not recommend that they be used as a proxy for a general notion of embedding quality.

	dev	test	p-value
Baseline	94.18	93.78	0.000
Rand. Proj.	94.33	93.90	0.006
GloVe	94.28	93.93	0.015
H-PCA	94.48	93.96	0.029
C&W	**94.53**	**94.12**	
CBOW	94.32	93.93	0.012
TSCCA	**94.53**	94.09	0.357

Table 4: F1 chunking results using different word embeddings as features. The p-values are with respect to the best performing method.

	test	p-value
BOW (baseline)	88.90	$7.45{\cdot}10^{-14}$
Rand. Proj.	62.95	$7.47{\cdot}10^{-12}$
GloVe	74.87	$5.00{\cdot}10^{-2}$
H-PCA	69.45	$6.06{\cdot}10^{-11}$
C&W	72.37	$1.29{\cdot}10^{-7}$
CBOW	**75.78**	
TSCCA	75.02	$7.28{\cdot}10^{-4}$

Table 5: F1 sentiment analysis results using different word embeddings as features. The p-values are with respect to the best performing embedding.

Noun phrase chunking. First we use a noun phrase chunking task similar to that used by Turian et al. (2010). The only difference is that we normalize all word vectors to unit length, rather than scaling them with some custom factor, before giving them to the conditional random field (CRF) model as input. We expect that this task will be more sensitive to syntactic information than to semantic information.

Sentiment classification. Second we use a recently released dataset for binary sentiment classification by Maas et al. (2011). The dataset contains 50K movie reviews with a balanced distribution of binary polarity labels. We evaluate the relative performance of word embeddings at this task as follows: we generate embedding-only features for each review by computing a linear combination of word embeddings weighted by the number of times that the word appeared in the review (using the same bag-of-words features as Maas et al. (2011)). A LIBLINEAR logistic regression model (Fan et al., 2008) with the default parameters is trained and evaluated using 10 fold cross-validation. A vanilla bag of words feature set is the baseline (denoted as BOW here). We expect that this task will be more sensitive to semantic information than syntactic information.

Results. Table 4 shows the average F1-scores for the chunking task. The p-values were computed using randomization (Yeh, 2000) on the sentence level. First, we can observe that adding word vectors as features results in performance lifts with all embeddings when compared to the baseline. The performance of C&W and TSCCA is statistically not significant, and C&W does better than all the remaining methods at the $p = 0.05$ level. Surprisingly, although the performance of Random Projections is still last, the gap to GloVe and CBOW is now very small. Table 5 shows results on the sentiment analysis task. We recover a similar order of embeddings as in the absolute intrinsic evaluation, however, the order of TSCCA and GloVe is now reversed.

Discussion. Performance on downstream tasks is not consistent across tasks, and may not be consistent with intrinsic evaluations. Comparing performance across tasks may provide insight into the information encoded by an embedding, but we should not expect any specific task to act as a proxy for abstract quality. Furthermore, if good downstream performance is really the goal of an embedding, we recommend that embeddings be trained specifically to optimize a specific objective (Lebret and Collobert, 2014).

6 Discussion

We find consistent differences between word embeddings, despite the fact that they are operating on the same input data and optimizing arguably very similar objective functions (Pennington et al., 2014; Levy and Goldberg, 2014). Recent work suggests that many apparent performance differences on specific tasks are due to a lack of hyperparameter optimization (Levy et al., 2015a). Different algorithms are, in fact, encoding surprisingly different information that may or may not align with our desired use cases. For example, we find that embeddings encode differing degrees of information about word frequency, even after length normalization. This result is surprising for two reasons. First, many algorithms reserve distinct "intercept" parameters to absorb frequency-based effects. Second, we expect that the geometry of the embedding space will be primar-

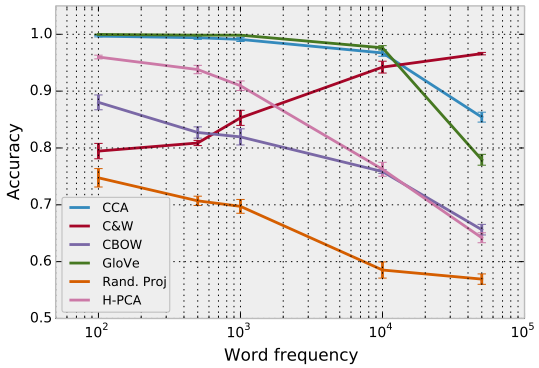


Figure 3: Embeddings can accurately predict whether a word is frequent or rare.

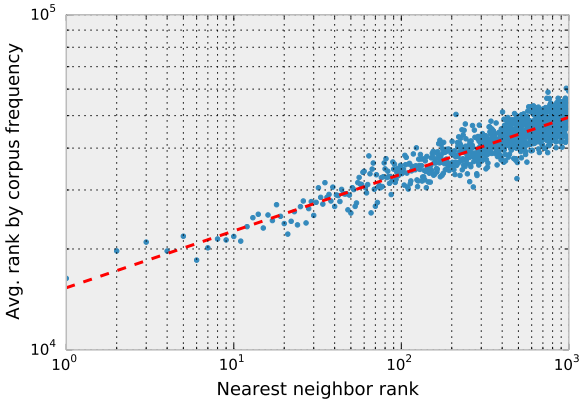


Figure 4: Avg. word rank by frequency in training corpus vs. nearest-neighbor rank in the C&W embedding space.

ily driven by semantics: the relatively small number of frequent words should be evenly distributed through the space, while large numbers of rare, specific words should cluster around related, but more frequent, words.

We trained a logistic regression model to predict word frequency categories based on word vectors. The linear classifier was trained to put words either in a frequent or rare category, with thresholds varying from 100 to 50,000. At each threshold frequency, we sampled the training sets to ensure a consistent balance of the label distribution across all frequencies. We used length-normalized embeddings, as rare words might have shorter vectors resulting from fewer updates during training (Turian et al., 2010). We report the mean accuracy and standard deviation (1σ) using five-fold cross-validation at each threshold frequency in Figure 3.

All word embeddings do better than random, suggesting that they contain some frequency information. GloVe and TSCCA achieve nearly 100% accuracy on thresholds up to 1000. Unlike all other embeddings, accuracy for C&W embeddings increases for larger threshold values. Further investigation revealed that the weight vector direction changes gradually with the threshold frequency — indicating that frequency seems to be encoded in a smooth way in the embedding space.

Although GloVe and CBOW are the two best performing embeddings on the intrinsic tasks, they differ vastly in the amount of frequency information they encode. As a consequence, we can conclude that most of the differences in frequency prediction are not due to intrinsic properties of natural language: it is not the case that frequent words naturally have only frequent neighbors.

Word frequency information in the embedding space also affects cosine similarity. For each of the words in the WordSim-353 dataset, we queried for the $k = 1000$ nearest neighbors. We then looked up their frequency ranks in the training corpus and averaged those ranks over all the query words. We found a strong correlation between the frequency of a word and its position in the ranking of nearest neighbors in our experiments. Figure 4 shows a power law relationship for C&W embeddings between a word's nearest neighbor rank (w.r.t. a query) and the word's frequency rank in the training corpus (nn-rank $\sim$ 1000 $\cdot$ corpus-rank$^{0.17}$). This is a concern: the frequency of a word in the language plays a critical role in word processing of humans as well (Cattell, 1886). As a consequence, we need to explicitly consider word frequency as a factor in the experiment design. Also, the above results mean that the commonly-used cosine similarity in the embedding space for the intrinsic tasks gets polluted by frequency-based effects. We believe that further research should address how to better measure linguistic relationships between words in the embedding space, e.g., by learning a custom metric.

7 Related work

Mikolov et al. (2013b) demonstrate that certain linguistic regularities exist in the embedding space. The authors show that by doing simple vector arithmetic in the embedding space, one can solve various syntactic and semantic analogy tasks. This is different to previous work, which phrased the analogy task as a classification problem (Turney, 2008). Surprisingly, word embed-

dings seem to capture even more complex linguistic properties. Chen et al. (2013) show that word embeddings even contain information about regional spellings (UK vs. US), noun gender and sentiment polarity.

Previous work in evaluation for word embeddings can be divided into intrinsic and extrinsic evaluations. Intrinsic evaluations measure the quality of word vectors by directly measuring correlation between semantic relatedness and geometric relatedness, usually through inventories of query terms. Focusing on intrinsic measures, Baroni et al. (2014) compare word embeddings against distributional word vectors on a variety of query inventories and tasks. Faruqui and Dyer (2014) provide a website that allows the automatic evaluation of embeddings on a number of query inventories. Gao et al. (2014) publish an improved query inventory for the analogical reasoning task. Finally, Tsvetkov et al. (2015) propose a new intrinsic measure that better correlates with extrinsic performance. However, all these evaluations are done on precollected inventories and mostly limited to local metrics like relatedness.

Extrinsic evaluations use embeddings as features in models for other tasks, such as semantic role labeling or part-of-speech tagging (Collobert et al., 2011), and improve the performance of existing systems (Turian et al., 2010). However, they have been less successful at other tasks such as parsing (Andreas and Klein, 2014).

More work has been done in unsupervised semantic modeling in the context of topic models. One example is the *word intrusion* task (Chang et al., 2009), in which annotators are asked to identify a random word inserted into the set of high probability words for a given topic. Word embeddings do not produce interpretable dimensions, so we cannot directly use this method, but we present a related task based on nearest neighbors. Manual evaluation is expensive and time-consuming, but other work establishes that automated evaluations can closely model human intuitions (Newman et al., 2010).

8 Conclusions

There are many factors that affect word embedding quality. Standard aggregate evaluations, while useful, do not present a complete or consistent picture. Factors such as word frequency play a significant and previously unacknowledged role. Word frequency also interferes with the commonly-used cosine similarity measure. We present a novel evaluation framework based on direct comparisons between embeddings that provides more fine-grained analysis and supports simple, crowdsourced relevance judgments. We also present a novel *Coherence* task that measures our intuition that neighborhoods in the embedding space should be semantically or syntactically related. We find that extrinsic evaluations, although useful for highlighting specific aspects of embedding performance, should not be used as a proxy for generic quality.

Acknowledgments

This research was funded in part through NSF Award IIS-1513692. We would like to thank Alexandra Schofield, Adith Swaminathan and all other members of the NLP seminar for their helpful feedback.

References

Jacob Andreas and Dan Klein. 2014. How much do word embeddings encode about syntax? In *ACL: Short Papers*, pages 822–827.

Javed Aslam and Mark Montague. 2001. Models for metasearch. In *SIGIR*, pages 276–284.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL*, pages 238–247.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *JAIR*, 49:1–47.

James McKeen Cattell. 1886. The time taken up by cerebral operations. *Mind,*, (42):220–242.

Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. In *NIPS*, pages 288–296.

Yanqing Chen, Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2013. The expressive power of word embeddings. *arXiv preprint: 1408.3456*.

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

Paramveer S. Dhillon, Jordan Rodu, Dean P. Foster, and Lyle H. Ungar. 2012. Two step CCA: A new spectral method for estimating vector models of words. In *ICML*, pages 1551–1558.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *JLMR*, 9:1871–1874.

Manaal Faruqui and Chris Dyer. 2014. Community evaluation and exchange of word vectors at wordvectors.org. In *ACL: System Demonstrations*.

Lev Finkelstein, Ehud Rivlin Zach Solan Gadi Wolfman Evgeniy Gabrilovich, Yossi Matias, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *TOIS*, 20(1):116–131, January.

Bin Gao, Jiang Bian, and Tie-Yan Liu. 2014. WordRep: A benchmark for research on learning word representations. *ICML Workshop on Knowledge-Powered Deep Learning for Text Mining*.

Rémi Lebret and Ronan Collobert. 2014. Word embeddings through Hellinger PCA. In *EACL*, pages 482–490.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *NIPS*, pages 2177–2185.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015a. Improving distributional similarity with lessons learned from word embeddings. *TACL*.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015b. Do supervised distributional methods really learn lexical inference relations? In *NAACL*.

Ping Li, Trevor J Hastie, and Kenneth W Church. 2006. Very sparse random projections. In *KDD*, pages 287–296.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *HLT-ACL*, pages 142–150.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL: System Demonstrations*, pages 55–60.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Tomas Mikolov, Wen-Tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

George A Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *HLT-NAACL*, pages 100–108.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *EMNLP*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *ACL*, pages 384–394.

Peter D. Turney. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *COLING*, pages 905–912.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *ACL*, pages 947–953.