Comparison of word embedding methods on intrinsic and extrinsic evaluations

Eunju Park(yello.ejp@gmail.com)

ABSTRACT

This paper compares abilities of three word embeddings which are distribution-based methods. From experiment of intrinsic and extrinsic evaluations, it is noticeable that there is no common method achieving better performance in all evaluation tasks. The experiments and results underline once again the fact that there is no grounded method to evaluate abilities of word embeddings. Until an effective evaluation method is proved, additional experiments on real world datasets from different domains should be encouraged in order to ensure the abilities.

1. INTRODUCTION

Many methods have been introduced to present features and contents of a document in Natural Language Processing tasks. Especially, vector space models for word representations have succeeded in capturing semantics and syntactics of languages. These methods are powerful tools widely used in NLP tasks, which can be applied to various applications such as information retrieval, document classification, question answering, named entity recognition and so on. Word embedding methods for word representations can be categorized into two parts depending on its approach: Count-based method and Distribution-based method. According to several experiments on intrinsic evaluations, distribution-based methods such as Skip-gram and CBOW(Continuous Bag-Of-Words) seem to generate better performance than count-based methods. In this paper, it compares abilities of 3 different distribution-based word embeddings on intrinsic and extrinsic evaluations. It trains word embeddings with a corpus collected by 18 participants under two popular word embedding methods, Skip-gram and CBOW, and uses word embeddings pretrained on Google News. Instead of using datasets for word similarity task which is one of intrinsic evaluations, it creates a set of 200 word pairs by using WordNet. For extrinsic evaluations, it uses the corpus for text classification. Then, it evaluates the abilities of each embedding method on both evaluations.

2. RELATED WORKS

2.1. Word embedding methods

1) Count-based method

The idea of count-based methods come from that the frequency of each word can be features representing statistical information of a document. In count-based methods, a word is represented by co-occurrence of words in contexts. Many baseline methods have been introduced such as LSA(Latent Semantic Analysis) and HAL(Hyperspace Analogues to Language) which are believed to be fast in training. LSA builds word-document matrix where rows correspond to words, columns correspond to documents and entries correspond to the number of occurrences of each word. HAL builds word-word matrix where rows and columns correspond to words and entries correspond to the number of occurrences of a given word occured in the context of another given word. However, the shortcoming of LSA and HAL is that the most frequent words have bigger weights comparing to other less frequent words causing a disproportionate. To address the shortcoming, normalized methods such as PPMI(Positive Pointwise Mutual Information) and HPCA(Hellinger PCA) have been suggested. By decomposing the original embedding matrix into lower dimensionality matrix, the data sparsity can be reduced.

2) Distribution-based method

The idea of distribution-based methods started from averaging random vectors. It believes that a representation of a word can be predicted from its neighbors and optimized. Many methods have been introduced such as Skip-gram and CBOW by utilizing given words and word's context. Skip-gram method predicts a word's context given the word itself while CBOW(Continuous Bag-Of-Words) method predicts a word given its context. In CBOW, each word is represented by random vectors, e.g., one-hot-vector, and the word's context is represented by the sum of vectors of words within the context window. At the end, a classifier predicts the target word. Skip-gram model is the other way around of CBOW model. As mentioned earlier, distribution-based methods are better on word analogy tasks and faster than count-based methods to train.

2.2. Evaluation methods

There are two mechanisms to evaluate abilities of word embeddings: intrinsic evaluation and extrinsic evaluation. Intrinsic evaluation uses sets of words labeled manually by experts and compares judged datasets with word embeddings. On the other hand, extrinsic evaluations pay attention to the ability of word embeddings to be used in various downstream NLP tasks such as named entity recognition, sentiment analysis, part-of-speech tagging, text classification, metaphor detection, etc.

3. EXPERIMENT FRAMEWORK

This paper focuses on Skip-gram, CBOW and pretrained embeddings on Google News. Table 1 shows the number of citations of recent embedding methods. It seems that distribution-based methods are more popular than count-based methods. Table 2 shows a brief description of pretrained word embeddings. With the assumption that the larger corpus helps to improve the performance, the main goal is to compare Word2Vec and the performances depending on the size of the corpus.

Category	Method	Citation
Count-based method	PPMI	721
	HPCA	238
Distribution-based method	**Skip-gram**	**17173**
	CBOW	
	FastText	3951

Table 1. The number of citations of recent embedding methods

Category of pretrained vectors	Data	Size
Word2vec **(distribution-based)**	**Google News**	**3 million words**
FastText (distribution-based)	Wikipedia	1 million words
	Common crawl	2 million words
Glove (count and distribution-based)	Wikipedia	400K words
	Common crawl	2.2 million words

Table 2. Pretrained word embeddings

3.1. Data

To train word embeddings by Skip-gram and CBOW models, it uses the corpus collected by 18 participants. The corpus has 658902 sentences and the number of tokens is 14548903. Instead of using all words occurred in the corpus, it requires to preprocess the data in order to remove some words. According to Zipf's law, some words occurred frequently in a corpus might not be meaningful in NLP tasks. For example, articles and prepositions occur frequently but do not present features and contents of a document. As counting the frequency of each word, it is recognized that the number of occurrences of top 100 frequent words is 7292670 and it covers 50% of occurrences. In this paper, only words which number of occurrences is between 30 and 19000 times are considered.

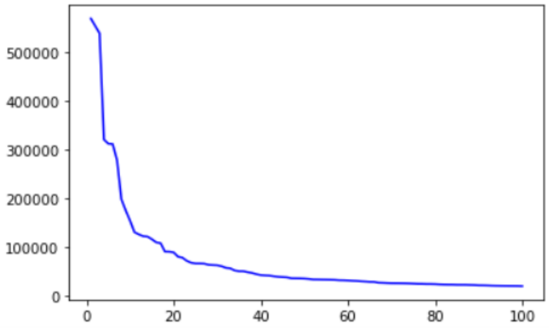

Figure 1. Number of words vs. number of occurrences per word

3.2. Word pairs for word similarity

There are many datasets of word pairs for intrinsic evaluations such as WS-353, SimVerb-3500, SimLex, etc. Instead of using existing datasets, it creates a set of 200 word pairs by using an online dictionary called WordNet. It provides several meanings of a word through Synsets, which provides 4 different word classes: noun, verb, adverb and adjective. Comparing a class to the other classes does not make sense. In order to achieve numerous pairs, only nouns are considered.

Word class	Number of Synsets
Noun	**119034**
Verb	11531
Adverb	4481
Adjective	21538

Table 3. The number of meaning for word classes

Through Synsets, calculating a distance between two words is possible by using semantic relations defined in Synsets. It provides several similarity measures such as path similarity, Wu-Palmer similarity, Leacock-Chodorow similarity, etc. Path similarity is straightforward to interpret the score but it is problematic when the words are in detailed levels. Among similarity measures, Wu-Palmer similarity is easy to interpret the score than others because the score is in the range from 0 to 1. The closer two words are, the closer score is to 1. If there is no semantic relation between two words, the score is equal to 0.

In WordNet, each word has more than one meaning. To compute similarity between two words, it averages similarities for all word senses. For example, the word *Boat* has 2 meanings and the word *Ship* has 1 meaning. It computes and averages two similarities between the word *Ship* and two senses on the word *Boat*.

Word	Synsets
Boat	Synset('boat.n.01'), Synset('gravy_boat.n.01')
Ship	Synset('ship.n.01')

Table 4. Number of senses on word *Boat* and *Ship* in WordNet

$$Sim(Boat, Ship) = \frac{Sim(1st\ Boat, Ship) + Sim(2nd\ Boat, Ship)}{2}$$

$$= \frac{0.9090+0.6000}{2} = 0.7545$$

After calculating similarities for 24082208 noun pairs, it randomly samples 100 pairs which similarities are between 0 and 0.5. And it repeats sampling on the pairs which range is from 0.5 to 1. Table 5 shows samples of word pairs in a set of 200 pairs.

Word1	Word2	Similarity
Scotland	Detroit	0.7000
panty	Classroom	0.5000
Benny	Einstein	0.6007
ordeal	Keeping	0.5242

Table 5. Samples of word pairs

3.3. Labels of sentences for text classification

It requires several steps to set a text classification task. The goal of text classification is to predict labels of sentences or documents. If there is a review on *Harry Potter and the Philosopher's Stone*, the category of the review would belong to Books. Training machine learning algorithms, a model can predict whether the review belongs to Books. Although the corpus does not have labels for each sentence unfortunately, classes for each sentence can be labeled depending on the corpus contexts. One of obvious label is to classify sentences by participants. As applying word embeddings to sentences, a sentence can be represented with sequences of vectors. However, presenting a sentence with sequences of vectors is too long because the dimension of a word is 300. Instead, it applies the concept of Bag-of Vocabulary. In Bag-of-Vocabulary, each word is unique so a sentence can be written with sequences of keys of words. After processing, it is ready to train a model through LSTM networks. Table 6 and 7 are examples describing how the processing works.

key	Word	key	Word
0	.	4	the
1	love	5	I
2	problem	6	all
3	is	7	that

Table 6. An example of Bag-of-Vocabulary

Original sentence	After processing	
	sentence	label
I love all.	[5 1 6 0]	Participant A
The problem is that.	[4 2 3 7 0]	Participant B

Table 7. Sentences after processing

4. EXPERIMENT AND RESULT

4.1. Word embeddings

It trains word embeddings on the corpus with Skip-gram and CBOW and uses pretrained word embeddings on Google News. The parameters of pretrained word embeddings such as the number of iterations, window size, etc. have not been known.

Type	Experiment setting	Dimension	Size
Skip-gram	Iteration=30, Window=5	300	1,5M words
CBOW	Iteration= 30, Window=5	300	1.5M words
Pre-trained	Unknown	300	3M words

Table 8. Description of word embeddings

Depending on methods, a word is represented with different values. For instance, the word *love* is represented like following table 9 in each word embedding.

Type	Representation
Skip-gram	-0.05764098 -0.05651434 -0.02088199 0.05422454 -0.05084962 …
CBOW	0.10642054 -0.01309693 -0.04140827 0.00130632 0.02166286 …
Pre-trained	0.10302734 -0.15234375 0.02587891 0.16503906 -0.16503906 …

Table 9. representation for word love

4.2. Intrinsic evaluation

For intrinsic evaluation, it creates 200 noun pairs in section 3.2. In vector space models, a similarity between two words can be obtained by computing the cosine similarity. By computing cosine similarities of word pairs listed in the set of noun pairs, it calculates the correlation for each embedding method. According to the results in table 10, it seems that CBOW achieves better performance than other methods.

Type	Correlation
Skip-gram	0.3042
CBOW	0.4031
Pre-trained	0.3882

Table 10. Correlation for each embedding method

4.3. Extrinsic evaluation

For the fair comparison, it sets the same parameters to train LSTM networks like table 11.

Parameters	Value
Number of training data	479830
Number of testing data	53316
Number of training iterations(epochs)	100
Optimizer	Adam
Loss	Cross Entropy

Table 11. Initial setting for LSTM networks

According to the results in table 12, it seems that there is no big difference in performances. But, Skip-gram slightly outperforms other methods on training and testing data.

Type	Accuracy on training data	Accuracy on testing data
Skip-gram	0.8209	0.8218
CBOW	0.7929	0.8002
Pre-trained	0.8040	0.8060

Table 12. Accuracy of embedding methods

5. DISCUSSION

Unlike the general expectation that training word embeddings with a large size of corpus would help to improve the performance, CBOW is better on word similarity task and Skip-gram achieves higher accuracy on text classification task. In spite of the results, several things should be considered before concluding abilities of word embeddings.

1. No grounded standard for evaluations: training word representations is unsupervised learning. In table 9, a word *love* gets different vector spaces depending on methods. In spite of different values, three methods achieve similar performances on intrinsic and extrinsic evaluations.

2. Lack of intrinsic evaluations. Apart from the word similarity task, there are other evaluation methods: word analogy, thematic fit, concept categorization, synonym detection, outlier word detection, etc. Among many evaluation methods, it evaluates the embedding methods only on word similarity task.

3. Weakness of WordNet. There is a criticism saying that WordNet is too systematic and the hierarchical structure of word meanings is determined by human judgement. Due to these weaknesses, using WordNet to create a set of word pairs for intrinsic evaluations is not appropriate.

4. Weakness of intrinsic evaluations. Intrinsic evaluations use collections of word pairs with human judgements which are subjective. It changes a set of word pairs 3 times for further experiments. In each experiment, different embedding methods show better performances. As a result, it is not convincing to conclude that word similarity task is an effective evaluation to determine whether it is a better embedding method.

Attempt	Skip-gram	CBOW	Pretrained
1st	0.2310	**0.3118**	0.3012
2nd	**0.2769**	0.2529	0.2616
3rd	0.1528	0.3188	**0.3655**

Table 13. Correlation on different datasets

5. Absence of corpus analysis. For proper setting, the corpus analysis should be done. For example, it might be difficult to figure out the distinct differences between a collection of academic articles on COVID-19 collected by Tanvi Vishwas Joglekar and another collection of news on COVID-19 collected by Patrick Schedlbauer in terms of terminologies. To get precise predictions, considering other perspectives such as length of a document might be helpful to improve performances rather than considering only words. Moreover, the analysis can effect on which word embedding method to use.

6. Lack of extrinsic evaluations. Apart from the text classification task, there are other evaluation methods: named entity recognition, sentiment analysis, text classification, textual entailment detection, paraphrase detection, metaphor detection and so on. Among many evaluation methods, it evaluates the embedding methods only on text classification task. It seems not to be convincing concluding which embedding method outperforms better than other methods only with a result of text classification.

7. Out-of-Vocabulary. To solve the tasks, it uses the corpus to train word embeddings and pretrained word embeddings on Google News. It means that only words existing in the bag-of-word can be dealt with. If a document has a word not in the bag, it is hard to process the unseen word. Although Skip-gram and CBOW are popular methods, these are not suitable methods on online processing if encounting unseen words happens often. In this case, other embedding methods such as character level embeddings can be considered.

8. Absence of correlation between intrinsic and extrinsic evaluations. The primary goal of intrinsic evaluations is to provide insights into word embedding to apply it for applications. However, the relation between two different evaluation approaches has not found yet.

6. CONCLUSION

From the experiment results and discussions, the biggest issue in word embeddings is that there is no consensus on what the best way is to evaluate word embeddings. Until a solution is found for grounded evaluation, word embedding methods should be evaluated how effectively it works well on task specific NLP problems.

7. REFERENCE

[1] Faruqui, Manaal, et al. "Problems with evaluation of word embeddings using word similarity tasks." arXiv preprint arXiv:1605.02276 (2016).

[2] Chiu, Billy, Anna Korhonen, and Sampo Pyysalo. "Intrinsic evaluation of word vectors fails to predict extrinsic performance." Proceedings of the 1st workshop on evaluating vector-space representations for NLP. 2016.

[3] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[4] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[5] Bakarov, Amir. "A survey of word embeddings evaluation methods." arXiv preprint arXiv:1801.09536 (2018).

[6] Ulrich Heid, Christian Wartena, Johannes Schäfer. "Natural Language Processing: Computing Meaming." Stiftung Universität Hildesheim, Hildesheim, DE. 2020. Inclass Lecture.