COMP6211 Final Project Report

A comparison among different word embeddings methods

(Survey)

Group No: 6

XIAO Wenyi LI Zheng Zeng Ziqian

Abstract

Continuous representation of words in a vector space, or word embedding, has been proven to have better performance in natural language processing tasks such as word similarity, analogy and parsing. Recent approach to make word vectors could be concluded into two main parts: count-based method and predict-based method. Both of them follow the idea that words with similar distributional contexts should be close to each other in vector space to embed the words. In this report, we do a survey on different word embedding method: word2vec, GloVe, Swivel and FastText, training on text8 dataset and making evaluation on these methods in both similarity tasks and analogy tasks with various evaluation datasets.

Keywords: word embedding, word2vec

1. Introduction

Regarding deep learning in natural language processing, especially machine reading and comprehension, the basic idea is to encode the documents into vectors containing full information about the text, that is to learn distributed representations(embeddings) for words. In recent years, there are a number of approaches to focusing on word representation in order to improve the accuracy in reading and comprehension task. Methods which have received high attention are word2vec [1] (Skip-gram and CBOW) and GloVe [2]. Also there are some works, like Swivel [3] focused on optimizing the drawbacks of these two models and FastText [4] takes consideration on the internal structure of words. These methods outperformed other previous models on word analogy, word similarity, and named entity recognition tasks. In this report, we do a survey on different word embedding method: word2vec, GloVe, Swivel and FastText, training on text8 dataset and making evaluation on these methods in both similarity tasks and analogy tasks with various evaluation datasets.

2. Models

2.1. word2vec

The basic ideas of word2vec could be described as: first, words with similar distributional contexts ought to be close to each other in the embedding space; second, manipulating the distributional context should lead to similar translation in the embedding space. For example, the embedded representation of the word queen can be translated from the representation of king, man and woman:

$$\text{queen} \approx \text{king} - \text{man} + \text{woman}$$

In word2vec, it proposes two new model architectures for learning distributed representations of words that try to minimize computational complexity: CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word. Model shows in Figure 2.1. and the basic formula shows in Figure 2.2.

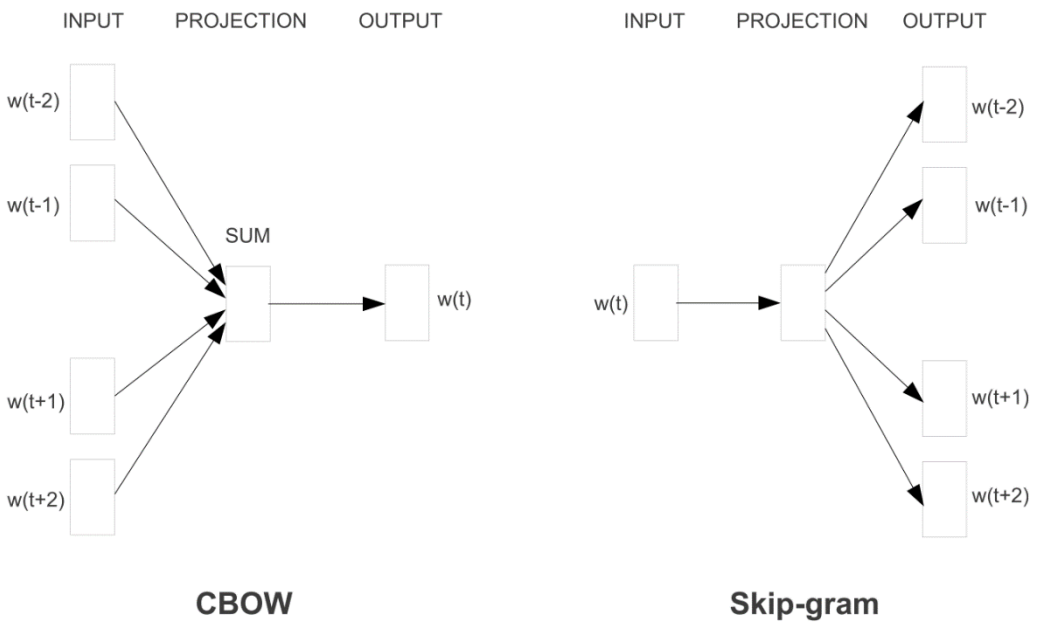

Figure 2.1. Models of CBOW and Skip-gram

$$\mathcal{L} = \sum_{w \in \mathcal{C}} \log p(w|Context(w)) \qquad \mathcal{L} = \sum_{w \in \mathcal{C}} \log p(Context(w)|w)$$

CBOW **Skip-gram**

Figure 2.2 Basic formulas of CBOW and Skip-gram

In most training works of word embedding, people use Skip-gram negative sampling, a method based on the Skip-gram model by minimizing the dot product between the focus word and a randomly sampled non-context word. The goal of it is to make an update to the embedding parameters θ to improve the following objective function:

$$J_{NEG} = \log Q_\theta(D = 1|w_t, t) + k\mathbb{E}_{\tilde{w} \sim P_{noise}}[\log Q_\theta(D = 0|\tilde{w}, h)]$$

It has been proved that SGNS could be regarded as implicit matrix factorization and the cells in this matrix are the point wise mutual information (PMI) of the respective word and context pairs as well as shifted by a global constant[5]. Point-wise mutual information (PMI) is a measure of co-occurrence between two features, defined as follows:

$$\mathrm{pmi}(x, y) = \log \frac{P(x,y)}{P(x)P(y)}$$

The SGNS model has outstanding performance on the analogy task compared with other traditional label-tagged methods, but they poorly utilize the statistics of the corpus since they train on separate local context windows instead of on global co-occurrence counts. In addition, it neglects the sequence the words and contexts occur in the text.

2.2. Global Vectors for Word Representation (GloVe)

GloVe is a global log-bilinear regression model that combines the advantages of the two major model families in the literature: global matrix factorization, such as latent semantic analysis (LSA) and local context window methods, such as the Skip-gram model. This model efficiently leverages statistical information by training only on the nonzero elements in a word-word co-occurrence matrix, rather than on the entire sparse matrix or on individual context windows in a large corpus. In addition, it has considered to preserve the linear directions of meaning, in other word, the sequence of words occurring in sentence. Using stochastic gradient descent, GloVe learns the model parameters for word embedding matrix W and context embedding matrix $\widetilde{W}$, also bias terms b_i and b_j. Therefore, the goal of GloVe is to minimize the following cost function:

$$\mathcal{L}_{Glove} = \sum_{i,j} f(x_{i,j})(w_i^T \widetilde{w}_j - \log x_{ij} + b_i + b_j)$$

Compared to SGNS which requires training time proportional to the whole size of the corpus, GloVe only require training time proportional to the number of observed co-occurrence pairs, shortening the training time.

Indeed, GloVe has outperformed other previous models on word analogy, word similarity, and named entity recognition tasks. However, it trains only on the observed co-occurrence statistics, while giving no penalty for placing features near to one another whose co-occurrence has not been observed, thus resulting in poor estimates for uncommon features.

2.3. Swivel

To handle the drawbacks of SGNS and GloVe, recently, another model for word embedding, Submatrix-wise Vector Embedding Learner (Swivel) used a method to combine factorization of PMI information matrix with a piecewise loss for making use of all the information in the matrix. Like Skip-gram, it makes use of the fact that many co-occurrence are unobserved; like GloVe, it works from co-occurrence statistics rather than by sampling; also like both, it performs a weighted

approximate matrix factorization of the PMI between features. In Swivel, it splits the whole co-occurrence matrix into shards after each rows and columns being sorted in descending order of feature frequency. In this way, it can parallelize the computation across many nodes at once. The partition process shows in Figure 2.3.

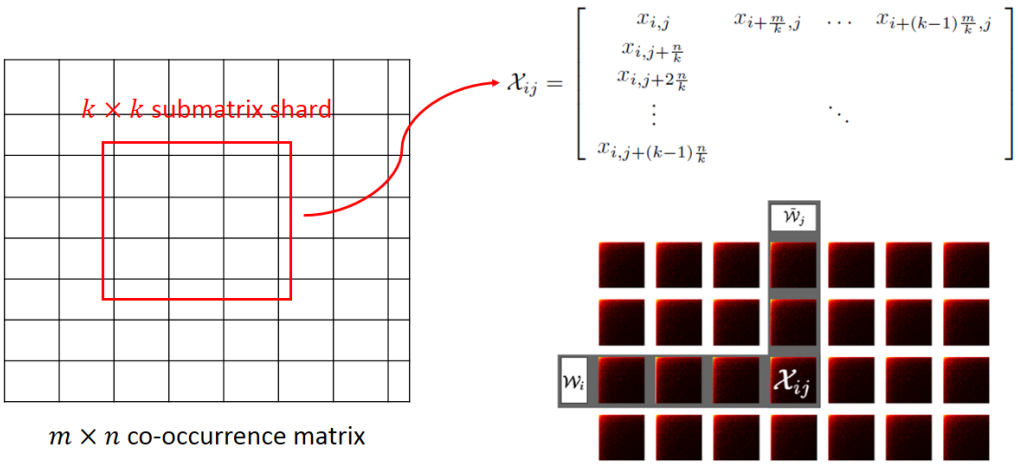

Figure 2.3 Partition to co-occurrence matrix in swivel

In training the model, swivel approximates the observed PMI of row feature i and feature j with $w_i^T \tilde{w}_j$. It computes the weighted squatted error between the embedding dot product and the PMI of feature i and feature j:

$$\mathcal{L}_{i,j} = \frac{1}{2} f(x_{ij}) \left(w_i^T \tilde{w}_j - pmi(i;j) \right)^2$$

However, when it comes to an unobserved co-occurrence pair, this model solved them simply by the requirement of avoiding over-estimating a smoothed PMI value with a "soft hinge", the light line in Figure 2.4.

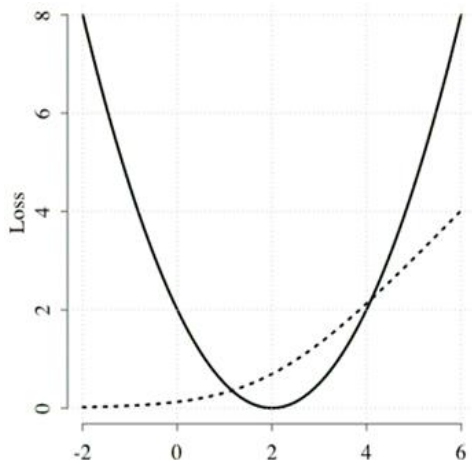

Figure 2.4: Loss as a function of predicted value of the embedding of two words.

This loss weighted function still does not given a more principled approach:

$$\log\left[1+\exp\left(w_i^T\widetilde{w}_j-pmi^*(i;j)\right)\right]$$

2.4. FastText

Another method which considers on the subword information across words is FastText. This character-level method, an extension of Skip-gram model, takes consideration on the internal structure of words, thus making connection between rare words and relatively frequent words. In this way, it allows to learn reliable representation for rare words. This model is a modified model of word2vec, also using local context to embed words in vector space. The details of FastText shows as follows:

(1) Given a word w, FastText denotes it by $\mathcal{G}_w \subset \{1,\dots,G\}$, the set of n-gram appearing in w. It represents a word by the sum of the vector representations of its n-gram, then getting the score by using the result to make dot product with the embedding of its context.

$$s(w,c)=\sum_{g\in\mathcal{G}_w} z_g^T v_c$$

(2) For the word at position t and the context c, the negative log-likelihood is:

$$\log\left(1+e^{-s(w_t,w_c)}\right)+\sum_{n\in\mathcal{N}_{t,c}}\log(1+e^{s(w_t,n)})$$

(3) Introducing loss function $\ell: x\mapsto\log(1+e^{-x})$, the objective is:

$$\sum_{t=1}^{T}\sum_{c\in C_t}\ell\left(s\left(w_t,\,w_c\right)\right)+\sum_{n\epsilon\mathcal{N}_{t,c}}\ell(-s(w_t,\,n))$$

The goal is to minimize the loss in the above function.

Compared to other methods of character-level word representation, this method does not need morphological segmenters. In other words, it does not require any preprocessing of the data, making it fast to apply. Since it shares the representations across words, it improves representation for rare words, linking them to common words in structural level.

However, this approach still has some shortcomings. When represent a word as a set of n-grams, the splitting process is relatively simple since it fixes the length of n of n-gram. While sometimes, the representation of a fixed segment of a word is meaningless, which cannot represent the prefixes or suffixes of the word.

3. Tasks

We evaluate the four words embedding models in two major tasks namely word similarity task and analogy task.

3.1. Word Similarity Task

The word similarity datasets were constructed by asking human subjects to rate the degree of semantic similarity or relatedness between two words on a numerical scale[6]. The performance is measured by the Pearson correlation of the two word embeddings' cosine distance and the average score given by the participants. The Pearson correlation of two variables can be calculated as follows:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

where: cov is the covariance and σ_X is the standard deviation of X.

3.2. Analogy Task

The analogy task is to evaluate the capacity of word embeddings to make an analogy like "man is to *woman* as king to *queen*" or "amazing is to amazingly as apparent is to apparently". There are two kind of analogy task, namely semantic analogy and syntactic analogy. "man is to *woman* as king to *queen*" is semantic analogy and amazing is to amazingly as apparent is to apparently" is syntactic analogy.

4. Experiment

4.1. Corpus

We train models on two corpuses, a 100M small one, named text8[1]. This corpus is extracted from English Wikipedia Dumps.

4.2. Evaluation Datasets

In word similarity task, we do experiment on 10 evaluation datasets. The information of 10 datasets[2] are shown in table 4.1 [7]. In analogy task, we do experiment on 14 kind of evaluation datasets, among which, 5 are semantic task and the rest are syntactic task [8]. The information of 14 datasets is shown in table 4.2.

No.	Corpus Name	Word Pairs	Reference
1	EN-MC-30	30	Miller and Charles, 1991
2	EN-WS-353-SIM	203	Agirre et. al, 2009
3	EN-MTurk-287	287	Radinsky et. al, 2011
4	EN-MTurk-771	771	Halawi and Dror, 2012
5	EN-WS-353-REL	252	Agirre et. al, 2009
6	EN-MEN-TR-3k	3000	Bruni et. al, 2012
7	EN-RW-STANFORD	2034	Luong et. al, 2013
8	EN-WS-353-ALL	353	Finkelstein et. al, 2002
9	EN-YP-130	130	Yang and Powers, 2006
10	EN-RG-65	65	R and G, 1965

Table 4.1 Word Similarity Evaluation Datasets Details

1 http://mattmahoney.net/dc/textdata.html

2 http://alfonseca.org/eng/research/wordsim353.html
http://tx.technion.ac.il/~kirar/Datasets.html
http://www2.mta.ac.il/~gideon/mturk771.html
http://clic.cimec.unitn.it/~elia.bruni/MEN.html
http://www-nlp.stanford.edu/~lmthang/morphoNLM/
http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/

No	Task Name	Number	Example
1	capital-common-countries	506	london england paris france
2	capital-world	3564	beijing china berlin germany
3	currency	596	usa dollar europe euro
4	city-in-state	2330	phoenix arizona seattle washington
5	family	420	boy girl brother sister
6	gram1-adjective-to-adverb	992	amazing amazingly apparent apparently
7	gram2-opposite	756	acceptable unacceptable aware unaware
8	gram3-comparative	1332	bad worse big bigger
9	gram4-superlative	992	bad worst big biggest
10	gram5-present-participle	1056	code coding dance dancing
11	gram6-nationality-adjective	1521	egypt egyptian china chinese
12	gram7-past-tense	1560	dancing danced decreasing decreased
13	gram8-plural	1332	banana bananas bird birds
14	gram9-plural-verbs	870	decrease decreases describe describes

Table 4.2 Word Analogy Evaluation Datasets Details

4.3. Parameters Setting

The dimension of word vector is set to 300 in all experiment.

4.3.1. The parameters setting in word2vec

Parameters	CBOW	Skip-gram
Learning rate	0.025	0.025
Window	5	5
Negative sample	5	5

4.3.2. The parameters setting in Glove

Parameters	Glove
Learning rate	0.025
window	5
alpha	0.75

4.3.3. The parameter settings in FastText

Parameters	CBOW	Skip-gram
Learning rate	0.025	0.025
Character n-gram	3	3
Negative sample	5	5

4.3.4. The parameter settings in Swivel

Parameters	Swivel
Learning rate	1
Shard size	4096×4096
Window size	10

4.4. Experiment Environment and Usage

CPU：Architecture: x86_64 CPU(s): 32
Model name: Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz
Memory：126G

During the experiment, we have found Swivel model will use more memory than other methods, since it split the training matrix into shards and every shard is trained in the same time.

	CPU	Memory
GloVe	100%	0.1%
Word2vec-bow	100%	0.1%
Word2vec-skip-gram	100%	0.1%
Swivel	**893%**	**8.2%**
FastText-cbow	100%	1.7%
FastText-skip-gram	100%	1.7%

4.5. Experiment results

4.5.1. Comparison on Word Similarity Task

We have compared all the methods on word similarity task. The details can be found in the Table 4.3 below. From the table, we can found word2vec-skip-gram model has achieved advanced results in most of types against other methods. However, Glove and Swivel never outperform other methods in all types.

	word2vec		**GloVe**	**Swivel**	**FastText**	
	CBOW	**Skip-gram**			**CBOW**	**Skip-gram**
EN-MC-30	65.60	65.78	44.77	56.07	**71.20**	47.29
EN-WS-353-SIM	71.95	**73.17**	52.11	72.03	62.57	65.39
EN-MTurk-287	63.83	**66.36**	51.51	62.81	63.87	64.90
EN-MTurk-771	59.84	**59.95**	43.80	52.98	56.70	59.18
EN-WS-353-REL	62.64	**67.11**	53.93	65.14	57.99	61.95
EN-MEN-TR-3k	64.28	67.54	44.25	59.18	63.47	**69.95**
EN-RW-STANFORD	37.77	36.06	24.54	34.54	**44.04**	39.69
EN-WS-353-ALL	67.87	**71.22**	48.80	68.50	60.97	63.32
EN-YP-130	34.08	39.67	33.20	25.00	33.63	**41.22**
EN-RG-65	65.47	**65.62**	37.25	50.32	64.43	57.53

Table 4.3: word similarity task results among different models on **text8** dataset

4.5.2. Comparison on Word Analogy Task

- **Semantic Task**
  We have compared all the methods on word analogy (semantic) task. The details can be found in the Table 4.4 below. From the table, we can found Swivel performs better than others in most of types, but Glove and FastText get bad performances.

	Word2vec		GloVe	Swivel	Fasttext	
	CBOW	Skip-gram			CBOW	Skip-gram
capital-common-countries	62.85	**70.95**	61.46	70.36	9.49	52.57
capital-world	24.49	32.97	26.04	**39.67**	1.88	18.13
currency	10.57	11.24	5.03	**12.58**	0	3.02
city-in-state	20.56	**34.42**	26.82	28.84	1.03	7.77
family	**55.71**	46.90	43.10	42.62	27.62	32.86
mean	26.52	35.06	27.98	**36.35**	3.44	16.84

Table 4.4: word analogy task(semantic) results

- **Syntactic Task**
  Finally, we have compared all the methods on word analogy (syntactic) task. The details can be found in the Table 4.5 below. From the table, we can found FastText performs better than others, but most methods like Glove, Swivel, word2vec-skip-gram perform very badly in all types.

	Word2vec		GloVe	Swivel	FastText	
	CBOW	Skip-gram			CBOW	Skip-gram
gram1-adjective-to-adverb	11.90	7.66	4.54	3.83	**69.96**	58.67
gram2-opposite	11.90	7.94	3.57	3.70	**77.25**	43.25
gram3-comparative	61.49	39.71	26.95	21.85	**63.14**	61.49
gram4-superlative	23.19	10.18	10.08	7.86	**74.50**	46.37
gram5-present-participle	31.34	14.39	14.11	11.84	**43.09**	29.92
gram6-nationality-adjective	66.34	79.16	57.59	72.39	66.14	**82.18**
gram7-past-tense	**32.44**	23.21	12.95	15.86	21.41	15.45
gram8-plural	43.39	38.44	25.60	40.24	74.02	**81.98**
gram9-plural-verbs	30.46	21.72	7.13	10.46	**81.26**	63.45
mean	37.90	30.59	20.76	24.53	**60.95**	54.16

Table 4.5: word analogy task(syntactic) results

4.6. Discussion

1. In similarity task, word2vec skip-gram performs better than others
2. In similarity task, Glove and Swivel never win in any dataset.
3. In analogy task (semantic), Swivel performs better than others.
4. In analogy task(semantic), Glove and Fasttext never win in any dataset.
5. In analogy task(syntactic), Fasttext performs better than others.
6. In analogy task(syntactic), Glove, Swivel, word2vec(skip-gram) never win.

5. Conclusion

In this report, we do a survey on different word embedding method: word2vec, GloVe, Swivel and FastText, training on text8 dataset and making evaluation on these methods in both similarity tasks and analogy tasks with various evaluation datasets. We found that in small dataset, or the circumstance that dataset are less available, skip-gram performs better in catching the meaning of words and FastText has an outstanding performance on learning the subword information or structural information of words. In the future, we plan to make a comparison among these method on large dataset, like full Wikipedia dump.

Reference

[1] Mikolov T, Dean J. Distributed representations of words and phrases and their compositionality[J]. Advances in neural information processing systems, 2013.
[2] Pennington J, Socher R, Manning C D. Glove: Global Vectors for Word Representation[C]//EMNLP. 2014, 14: 1532-43.
[3] Shazeer N, Doherty R, Evans C, et al. Swivel: Improving Embeddings by Noticing What's Missing[J]. arXiv preprint arXiv:1602.02215, 2016.
[4] Bojanowski P, Grave E, Joulin A, et al. Enriching Word Vectors with Subword Information[J]. arXiv preprint arXiv:1607.04606, 2016.
[5] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In Advances in Neural Information Processing Systems, pages 2177–2185, 2014.
[6] Miller, George A., Charles, Walter G. Contextual correlates of semantic similarity 1991
[7] D. Yang and D. M. Powers, Verb Similarity on the Taxonomy of WordNet. Masaryk University, 2006.
[8] Rubenstein, Herbert, Goodenough, John B. Contextual correlates of synonymy 1965