Improving Word Representations: A Sub-sampled Unigram Distribution for Negative Sampling

Abstract

Word2Vec is the most popular model for word representation and has been widely investigated in literature. However, its noise distribution for negative sampling is decided by empirical trials and the optimality has always been ignored. We suggest that the distribution is a sub-optimal choice, and propose to use a sub-sampled unigram distribution for better negative sampling. Our contributions include: (1) proposing the concept of semantics quantification and deriving a suitable sub-sampling rate for the proposed distribution adaptive to different training corpora; (2) demonstrating the advantages of our approach in both negative sampling and noise contrastive estimation by extensive evaluation tasks; and (3) proposing a semantics weighted model for the MSR sentence completion task, resulting in considerable improvements. Our work not only improves the quality of word vectors but also benefits current understanding of Word2Vec.

Introduction

The recent decade has witnessed the great success achieved by word representation in natural language processing (NLP). It proves to be an integral part of most other NLP tasks, in which words have to be vectorized before input to the models. High quality word vectors have boosted the performance of many tasks, such as named entity recognition BIBREF0, BIBREF1, sentence completion BIBREF2, BIBREF3, part-of-speech tagging BIBREF4, BIBREF5, sentiment analysis BIBREF6, BIBREF7, and machine translation BIBREF8, BIBREF9. In a conventional way, word vectors are obtained from word-context co-occurrence matrices by either cascading the row and column vectors BIBREF10 or applying singular value decomposition (SVD) BIBREF11. However, these approaches are limited by their sub-optimal linear structure of vector space and the highly increased memory requirement when confronting huge

vocabularies. Both problems have been solved by a popular model called Word2Vec BIBREF12, which utilizes two shallow neural networks, i.e., skip-gram and continuous bag-of-words, to learn word vectors from large corpora. The model is also capable of capturing interesting linear relationships between word vectors.

While Word2Vec makes a breakthrough in word representation, it has not been fully understood and its theoretical exploitation is still in demand. One aspect, which has always been ignored, is the choice of noise distribution for negative sampling. Word2Vec employs a smoothed unigram distribution with a power rate of 3/4 as the noise distribution. The decision is made by empirical trials but has been widely adopted in subsequent work BIBREF13, BIBREF4, BIBREF14, BIBREF15. However, the quality of learned word vectors is sensitive to the choice of noise distribution BIBREF16, BIBREF13 when using a moderate number (5 to 15) of negative samples, which is a common strategy for the tradeoff between vector quality and computation costs.

In this paper, we propose to employ a sub-sampled unigram distribution for negative sampling and demonstrate its capability of improving the linear relationships between word vectors. Our contributions include three aspects: (1) We propose the concept of semantics quantification and derive a suitable sub-sampling rate for the proposed distribution. (2) We demonstrate the advantages of our noise distribution in both negative sampling and noise contrastive estimation by extensive experiments. (3) We propose a semantics weighted model for the MSR sentence completion task, resulting in considerable improvements.

Word2Vec ::: Architectures

Firstly, we briefly introduce the two architectures, i.e., skip-gram (SG) and continuous bag-of-words (CBOW) in Word2Vec BIBREF12. For a corpus with a word sequence $w_{1}, w_{2}, \cdots , w_{T}$,

skip-gram predicts the context word w_{t+j} given the center word w_t, and maximizes the average log probability,

where c is the size of context window, and $p(w_{t+j}|w_{t})$ is defined by the full softmax function,

where v_{w} and $v_{w}^{\prime }$ are the vectors of the “input” and “output” words, and $|V|$ is the size of vocabulary.

As for CBOW, it predicts the center word based on the context words. The input vector is usually the average of the context words' vectors, i.e., $v_{w_{I}} = \frac{1}{2c} \sum _{-c \le j \le c, j \ne 0} v_{w_{t+j}}$.

Word2Vec ::: Negative Sampling

For large vocabularies, it is inefficient to compute the full softmax function in Eq. (DISPLAY_FORM3). To tackle this problem, Word2Vec utilizes negative sampling to distinguish the real output word from k noise words,

where $\sigma (x) = \frac{1}{1 + \exp (-x)}$, and P_n is the so-called noise distribution, representing the probability for a word to be sampled as a noise word. The smoothed unigram distribution used in Word2Vec is expressed as,

where $f(w_i)$ is the frequency of word w_i.

Word2Vec ::: Sub-sampling

Sub-sampling is a process in Word2Vec for randomly deleting the most frequent words during training, since they are usually stop words with less information than infrequent ones. During sub-sampling, the probability that a word w_i should be kept is defined as,

where $\hat{f}(w_i)$ is the normalized word frequency of w_i, and t is called the sub-sampling rate typically between 10^{-5} and 10^{-3}. The process does not delete infrequent words.

Related Work

Unigram. A noise distribution is recommended to be close to the distribution of the real data in noise contrastive estimation (NCE) BIBREF16. Such guidance finds its earliest application for training language models by BIBREF17, demonstrating that the unigram distribution works better than a uniform distribution. This choice is also adopted in some other work BIBREF18, BIBREF19, BIBREF20, BIBREF21. However, the performance of models is limited due to the inadequate training of infrequent words BIBREF22, BIBREF23.

Smoothed Unigram. The smoothed unigram distribution in Word2Vec BIBREF12 solves this problem because it gives more chances for infrequent words to be sampled. However, the required power rate is decided empirically, and may need adjustment for different scenarios BIBREF24, BIBREF25. BIBREF23 even propose to use a bigram distribution after studying the power rate, but it is infeasible for large corpora. Besides, the smoothed unigram distribution also changes the lexical structure of infrequent words, which could be a reason for the limited quality of word vectors.

Sub-sampled Unigram Distribution

We believe a sub-sampled unigram distribution is better for negative sampling since it reduces the

amount of frequent words and also maintains the lexical structure of infrequent words. To our best knowledge, we are the first to employ such a noise distribution for negative sampling. Beyond this, we propose a approach to derive the sub-sampling rate that is adaptive to different corpora (Table TABREF35).

Sub-sampled Unigram Distribution ::: Critical Word

We start our analysis by recalling the probability in Eq. (DISPLAY_FORM9) of a word to be kept during sub-sampling. Obviously, we need to choose the sub-sampling rate t to decide the noise distribution. Although empirically selecting a sub-sampling rate can result in improvements (Table TABREF55), we aim to derive the sub-sampling rate adaptive to different corpora. To accomplish this, we firstly introduce a concept critical word denoted by w_{crt}, which is the word with $P_{keep}(w_{crt})=1$. The critical word indicates that words with frequencies lower than it will not be deleted during sub-sampling. It is uniquely decided by the sub-sampling rate. Thus, if we select the critical word with certain properties at first, we are able to obtain a suitable sub-sampling rate in return.

The basic rule for us to select the critical word is to find a word with balanced semantic and syntactic information. We prefer not to delete words with relatively more semantic information. Now, the problem is how to measure these two kinds of information a word possesses.

Sub-sampled Unigram Distribution ::: Semantics Quantification

In order to quantify the semantic and syntactic information of words, we consider two observations: (1) frequent words are more likely to be function words with more syntactic information; (2) infrequent words are more likely to be content words with more semantic information BIBREF26. Thus, for the r-th most frequent word w, the quantity of its semantic and syntactic information I_{sem}^{w} and

I_{syn}^{w}, can be described as,

where $F_1(r)$ and $F_2(f_r)$ are monotonically increasing functions of the ranking r and the frequency f_r, respectively. One can tell that the functions capture the properties of the observations.

On the other hand, we require that the total quantity of semantic and syntactic information, denoted by I_{tot}^{w} is fixed for all words, i.e.,

where $\mathrm {const}_1$ is a constant. We rewrite Eq. (DISPLAY_FORM14) into an exponential form as the following,

This expression leads us to a well known power law called Zipf's law BIBREF27, which approximates the relationship between f_r and r as,

where $\gamma , \beta $ are constants and $\beta \approx 1$. Consequently, we can decide the form of the functions $F_1(r)$ and $F_2(f_r)$ as,

Obviously, the $\log $ form functions satisfy the definition we made before. As a results, the total information becomes $\log \gamma $ given $\beta \approx 1$.

Sub-sampled Unigram Distribution ::: Expression of Sub-sampling Rate

Now, given the quantified information, we are able to decide the critical word satisfying the condition

Combined with Eq. (DISPLAY_FORM16), we obtain the frequency of the critical word

where r_c is the ranking of the critical word. Meanwhile, we know the probability of the critical word w_{crt} to be kept should be exactly $P_{keep}^{t_c} (w_{crt})=1$. Thus, with Eq. (DISPLAY_FORM9) and Eq. (DISPLAY_FORM20), the sub-sampling rate for our noise distribution is expressed as

Note that we use t_c to distinguish from the sub-sampling rate t applied for the training corpus.

Sub-sampled Unigram Distribution ::: Constants Estimation

As for the estimation of constants $\gamma $ and $\beta $, we provide two choices:

(1) wLSE-1. We use weighted least squares estimation (wLSE) to estimate the two constants. Since more data are located at higher positions in $\log r$ axis, wLSE with a weight of $\frac{1}{r}$ for the r-th most frequent word makes sure the trend of line can be well fit. The estimated constants are

where $\left\langle x \right\rangle $ denotes the weighted average of x such that $\left\langle x \right\rangle = \sum _{r=1}^{|V|}\frac{x}{r} / \sum _{r=1}^{|V|}\frac{1}{r}$.

(2) wLSE-2. We use wLSE with a condition that the fitting line passes through the point $(\log 1, \log f_1)$. This method engages the most frequent word to further control the trend of the line. As a result, $\hat{\gamma }= f_1$ and

Now, we can write down the expression of the sub-sampled unigram distribution

where $\alpha _i$ satisfies

Note that we use P_n^{sub} to distinguish from the original noise distribution P_n in Word2Vec.

Sub-sampled Unigram Distribution ::: Discussions

In semantics quantification, the modeling of word distribution is not limited to zipf's law. We adopt it because of its popularity and conciseness. There could be other choices BIBREF28, BIBREF29, and the expression of t_c needs modification accordingly. Besides, one can either use the chosen law to decide the critical word or just search through the unigram distribution to find it.

Experiments

To show the advantages of our noise distribution, we conduct experiments on three evaluation tasks. While the word analogy task BIBREF12 is our focus for testing the linear relationships between word vectors, we also evaluate the learned word vectors on the word similarity task BIBREF0 and the synonym selection task BIBREF3.

In the following, we firstly describe the experimental setup including baselines, training corpora and details. Next, we report experimental results for the three NLP tasks. At last, we introduce the semantics weighted model proposed for the MSR sentence completion task BIBREF30.

Experiments ::: Experimental Setup ::: Baselines

We train the two models, SG and CBOW, using the original noise distribution and other two obtained by our approach, specifically,

(1) Uni$^{3/4}$. The smoothed unigram distribution proposed by BIBREF12.

(2) Sub^{L1}. The sub-sampled uinigram distribution, of which the threshold t_c is estimated by

wLSE-1.

(3) Sub^{L2}. The sub-sampled uinigram distribution, of which the threshold t_c is estimated by wLSE-2.

Experiments ::: Experimental Setup ::: Training Corpora

Our training corpora come from four sources, described as below:

(1) BWLM. The “One Billion Word Language Modeling Benchmark”, which is already pre-processed and has almost 1 billion tokens.

(2) Wiki10. The April 2010 snapshot of the Wikipedia corpus with a total of about 2 million articles and 1 billion tokens.

(3) UMBC. The UMBC WebBase corpus from the Stanford WebBase project’s February 2007 Web crawl, with over 3 billion tokens.

(4) MSR. The MSR corpus containing 5 Conan Doyle Sherlock Holmes novels with about 50 million tokens.

The first three large corpora are used for word similarity, synonym selection, and word analogy tasks. The MSR corpus is designated for the MSR sentence completion task. We pre-process the corpora by converting all words into lowercase and removing all the non-alphanumeric. The number of remaining tokens for each corpus is listed in the column Size of Table TABREF35. Vocabularies are built by discarding words whose occurrences are less than the threshold shown in the column Mcn. The column

Vocab represents the sizes of the resulted vocabularies. The rightmost two columns are the sub-sampling rates for our noise distribution by the wLSE-1 and wLSE-2 estimations, respectively. The values are 10^6 times of the true ones for readability.

Experiments ::: Experimental Setup ::: Training details

We implement the training of word vectors with the word2vec tool, in which the part of noise distribution is modified to support several choices. For SG and CBOW, we set the vector dimensionality to 100, and the size of the context window to 5. We choose 10 negative samples for each training sample in the models. The models are trained using the stochastic gradient decent (SGD) algorithm with a linear decaying learning rate with an initial value of 0.025 in SG and 0.05 in CBOW. We train the models on the three large corpora for 2 epochs, and for MSR's Holmes novels the value may vary. Results in this paper are shown in percentages and each of them is the average result of 4 repeated experiments, unless otherwise stated.

Experiments ::: Task 1: Word Similarity Task ::: Task Description

The task computes the correlation between the word similarity scores by human judgment and the word distances in vector space. We use Pearson correlation coefficient $\rho _p$ as the metric, the higher of which the better the word vectors are. The expression of $\rho _p$ is

where $\phi $ and $\hat{\phi }$ are random variables for the word similarity scores by human judgment and the cosine distances between word vectors, respectively. Benchmark datasets for this task include RG BIBREF31, MC BIBREF32, WS BIBREF33, MEN BIBREF34, and RW BIBREF35.

Experiments ::: Task 1: Word Similarity Task ::: Results

We implement the task on the mentioned 5 datasets and show the results in the column Word Similarity of Table TABREF42. At the first glance, our noise distributions Sub^{L1} and Sub^{L2} perform slightly better than Uni$^{3/4}$. Significant improvements can be achieved on two small datasets RG and MC, because they are more sensitive to the vector quality. Another observation is that CBOW is more affected by Sub^{L1} and Sub^{L2} than SG, if comparing results on RG and MC with Wiki10 corpus. These results show that our noise distributions have the potential as high as or even higher than the smoothed unigram distribution in learning good word vectors.

Experiments ::: Task 2: Synonym Selection Task ::: Task Description

This task attempts to select the semantically closest word, from the candidate answers, to the stem word. For example, given the stem word “costly” and the candidate answers “expensive, beautiful, popular, complicated”, the most similar word should be “expensive”. For each candidate answer, we compute the cosine similarity score between its word vector and that of the stem word. The candidate answer with the highest score is our final answer for a question. Here we use the TOEFL dataset BIBREF36 with 80 synonym questions and the LEX dataset with 303 questions collected by ourselves.

Experiments ::: Task 2: Synonym Selection Task ::: Results

We report the results of this task in the Synonym Selection column of Table TABREF42. For all the noise distributions, the results are not stable on TOEFL dataset since it is quite small. Still, Sub^{L1} and Sub^{L2} have comparable performance with Uni$^{3/4}$. In particular, Sub^{L1} makes considerable improvements with Wiki10 corpus. As for LEX dataset, Sub^{L1} and Sub^{L2} outperform Uni$^{3/4}$ in both SG and CBOW models with BWLM corpus. With the other two corpora, Sub^{L2} performs better than Sub^{L1} and Uni$^{3/4}$ using CBOW model. But again, the SG model appears to be less boosted by Sub^{L1} and Sub^{L2} in terms of the corresponding results.

Considering the unbalanced number of questions in these two datasets, we provide the total results on TOEFL+LEX and conclude that our noise distributions are better than Uni$^{3/4}$.

Experiments ::: Task 3: Word Analogy Task ::: Task Description

The task comes from the idea that arithmetic operations in a word vector space can be predicted: given three words w_a, w_b, and w_c, the goal is to find a word w_d such that the relation $w_d:w_c$ is the same as the relation $w_b:w_a$. Semantic questions are in the form of “Athens:Greece is as Berlin:German” and syntactic ones are like “dance:dancing is as fly:flying”. Here we choose the fourth word $\hat{w}_d$ by maximizing the cosine similarity such that $\hat{w}_d = \operatornamewithlimits{arg\,max}_{w\in V} \,\cos \left(v_{w_b}-v_{w_a}+v_{w_c}, v_w\right)$ BIBREF37. We test the learned word vectors on the Google analogy dataset BIBREF12, which contains 8,869 semantic questions and 10,675 syntactic ones.

Experiments ::: Task 3: Word Analogy Task ::: Results

This task is our primary focus because it exposes interesting linear relationships between word vectors. Thus we conduct four sub-experiments to investigate four aspects of our noise distributions.

Model Responses. The two models SG and CBOW respond differently to our noise distributions as shown in Table TABREF42. When applying CBOW model on the three corpora, our noise distributions Sub^{L1} and Sub^{L2} can result in significant improvements compared with Uni$^{3/4}$, especially on semantic questions. Specifically, the accuracy of semantic questions is improved by 2 to 6 points, and for syntactic questions it is 1.5 to 2 points. As for the SG model, the improvements on semantic questions by Sub^{L1} and Sub^{L2} are still considerable (2 to 5 points). But on syntactic questions, Uni$^{0.75}$ becomes competitive with Sub^{L1} and Sub^{L2} and is slightly better with BWLM and

Wiki10 corpora. The reason may be that SG model is better at capturing semantic relationships between words compared with CBOW model. Still, it is safe to say that our noise distributions are better for SG in terms of the total accuracy.

Number of Negative Samples. Increasing the number of negative samples does not reduce the advantages of our noise distributions necessarily. We report the results of the task using various number of negative samples in Fig. FIGREF48 (a) for CBOW and Fig. FIGREF48 (b) for SG. Note that we only train the models on Wiki10 and compare Sub^{L2} with Uni$^{3/4}$. For CBOW, Sub^{L2} outperforms Uni$^{3/4}$ consistently with significant margins on both semantic and syntactic questions. For SG, though the two distributions are competitive with each other on syntactic questions, Sub^{L2} always performs better than Uni$^{3/4}$ on semantic ones.

Optimality. Since our approach is built on assumptions and new concepts, we wonder whether the resulted t_c is optimal. We select several values around t_c-2 and show the word analogy results in Fig. FIGREF48 (c). For CBOW, t_c-2 approaches the optimal point given the accuracy on semantic questions and the total dataset. For SG, the optimal point lies between $0.1\,t_c$-2 and t_c-2, with negligible advantages relative to Sub^{L2}. Notice that the point $3.57\,t_c$-2 corresponds to 10^{-5}, showing much worse performance than Sub^{L2}. It indicates that trying a commonly used sub-sampling rate is inappropriate, and our approach is better.

Scalability. We apply our noise distributions in NCE, from which negative sampling originates, to train word vectors. The implementation comes from wang2vec by BIBREF4, and we report the results of this task using CBOW. We include the unigram distribution Uni BIBREF18 and the sub-sampled unigram distribution Sub^{1e-5} with a manually chosen threshold 10^{-5} for comparison. We draw three conclusions: (1) Uni$^{3/4}$ indeed works much better than Uni as claimed in BIBREF12; (2) Sub^{1e-5} results in considerable improvements compared with Uni$^{3/4}$, especially on semantic

questions; (3) Our Sub^{L2} achieves the best performance consistently even with a larger vector size of 300. Note that even though Sub^{1e-5} or Uni$^{3/4}$ performs better on syntactic questions with UMBC corpus, its results on semantic questions and the total dataset are much worse than Sub^{L2}. To this end, we believe that our approach is also scalable to the NCE related work.

Experiments ::: Extension of Semantics Quantification ::: MSR Sentence Completion Task

The task deals with incompletion sentences, e.g., “A few faint were gleaming in a violet sky.” with candidate answers “tragedies, stars, rumours, noises, explanations”, and aims to choose a word (e.g., “stars”) to best complete the sentence. Several works evaluate word vectors on this task BIBREF38, BIBREF18, BIBREF3 since it requires a combination of semantics and occasional logical reasoning. Most of them follow the same procedures of implementation described in BIBREF17. Specifically, we can calculate the probabilities that a set of words $\mathcal {S}$ surrounding the blank to be the context of each candidate answer w_{cd}. Then the score of the candidate answer is the sum of these probabilities,

and the highest score corresponds to the final answer for the question.

Since the conventional method ignores the syntactic structure of sentences, it should be biased to semantics. Thus, we modify the method with two steps: (1) applying sub-sampling on the words in the sentences (CM^s); and (2) using quantified semantics as weights to form a semantics weighted model (SWM) based on (1). Then we have

Experiments ::: Extension of Semantics Quantification ::: Results

The setup of models is a little different: the size of context window for SG and CBOW is 10 and 5; the

number of negative samples is 20 in both models; we train SG for 5 and 10 epochs when the size of word vectors is 100 and 300, while the number of epochs is 10 and 20 in CBOW; we use all the rest words in a sentence to form $\mathcal {S}$.

Our focus here is to popularize SWM rather than to compare the noise distributions. We show the results of this task by previous word presentation models and our approach in Table TABREF60. The bottom three previous models follow the conventional method. Accordingly, we draw two conclusions: (1) sub-sampling on the words in sentences results in significant improvements to the conventional method; and (2) SWM further improves CM^s and beats previous word representation models with a vector size of 300, indicating the success of semantics quantification.

Conclusions

We propose to employ a sub-sampled unigram distribution for better negative sampling, and design an approach to derive the required sub-sampling rate. Experimental results show that our noise distribution captures better linear relationships between words than the baselines. It adapts to different corpora and is scalable to NCE related work. The proposed semantics weighted model also achieves a success on the MSR sentence completion task. In summary, our work not only improves the quality of word vectors, but also sheds light on the understanding of Word2Vec.