INSTITUT
POLYTECHNIQUE
DE PARIS

Research paper reading review

A SIMPLE BUT TOUGH-TO-BEAT BASELINE FOR SENTENCE EMBEDDINGS

Work of:
Mohamed Issa

M2 Data Science

Academic year: 2021/2022

Contents

List of Figures

Introduction

A very important task in Natural language processing is retrieving information from corpus of text in order to capture the semantic of some word sequences and the existing similarities between words. This is the reason behind the development of different models (word2vec, Glove) computing word embeddings to map a given vocabulary into a representational continuous space in which the semantic and syntactic properties are quite preserved. This motivates the search for a representation for a given sentence by leveraging its word embeddings. An intuitive candidate is simply the average of the word embeddings of the sentence. In this article, we present a more accurate representation by associating specific weights to words constructing the sentence called the *(smooth inverse frequency)* **SIF**.

Unlike other weighting methods such as the TF-IDF, we show that this SIF method has a theoretical justification by applying it to the sentence generation model in [3] (Arora et al.). We show also that its used to explain some embedding methods like word2vec (CBOW) with subsampling and that this method overtakes over different baselines' methods and models. Further, we give an example where SIF weighting enhances performance on a given word embedding. Then we discuss the consistency of this theoretical setting.

1 Contribution to related works

1.1 Probability model of [2] (Arora et al.)

This paper work was intially motivated by the text generative model of the paper [2] The model of this paper produces words in a dynamic way. Given a discourse vector $c_t \in \mathbf{R}^d$ representing the context of a sentence, the probability of having the word w in the position t is given by the formula:

$$Pr(\text{w emitted at time t}/c_t) \propto exp(< c_t, v_w >) \tag{1}$$

where the $< .,. >$ denotes the inner product and v_w is the word vector for w taken from any word embedding.

The dynamic is described by a random walk, where the update of c_t to c_{t+1} is done by adding a random displacement vector in the unit sphere.

1.2 Improved probability model

By adapting the probability formula (1), it was shown in the paper [3] that the **MAP** estimate of the discourse vectors is just the average of the embeddings of the words in the sentence. The contribution of this paper lies in the new more realistic formulation of the probability given by:

$$Pr(\text{w emitted at time s}/c_s) = \alpha p(w) + (1-\alpha)\frac{exp(< \tilde{c}_s, v_w >)}{Z_{\tilde{c}_s}} \tag{2}$$

where $\tilde{c}_s = \beta c_0 + (1-\beta)c_s$, $c_0 \perp c_s$, α and β are hyperparameters, and $Z_{\tilde{c}_s}$ is a normalizing constant given by $Z_{\tilde{c}_s} = \sum_{w \in \mathcal{V}} exp(< \tilde{c}_s, v_w >)$.

This new model introduces two major new components:

- $\alpha p(w)$: The $p(w)$ denotes the unigram probability of the word w in the entire corpus. This term allows words out of context (having low inner product with $tildec_s$) to occur. Such words are like "the" and "and" that they don't show up only in a particular context.

- $\tilde{c}_s$: This term is composed of c_0 (which represents the common semantic of the sentences in the corpus) and c_s (which is the additive quantity presenting the context of the sentence). The c_0 can be approximated by the first principal component of the PCA fitted on the set of sentences. The c_t is then estimated by doing a projection on this first component and taking the residual value. This method can be seen as a denoising method.

1.3 Sentence embedding formula

The sentence embedding is defined by the maximum likelihood estimator of the vector c_s that generated the sentence s.Under the assumption that word embeddings are uniformly distributed, the $Z_{\tilde{c}_s}$ is set to a constant Z. Using the model 2, the likelihood of a sentence s is:

$$p[s/c_s] = \prod_{w \in s} p(w/c_s) = \prod_{w \in s} \left[\alpha p(w) + (1-\alpha) \frac{exp(< v_w, \tilde{c}_s >)}{Z} \right] \tag{3}$$

if we denote by $f_w(\tilde{c}_s) = log\left[\alpha p(w) + (1-\alpha)\frac{exp(<v_w,\tilde{c}_s>)}{Z}\right]$, the gradient of f_w w.r.t $\tilde{c}_s$ is:

$$\nabla f_w(\tilde{c}_s) = \frac{1}{\alpha p(w) + (1-\alpha)\frac{exp(<v_w,\tilde{c}_s>)}{Z}} \frac{1-\alpha}{Z} exp((< v_w, \tilde{c}_s >))v_w \tag{4}$$

Using Taylor expansion at 0, we can approximate f_w by:

$$\begin{aligned} f_w(\tilde{c}_s) &\approx f_w(0) + \nabla f_w(0)^T \tilde{c}_s \\ &= constant + \frac{(1-\alpha)/(\alpha Z)}{p(w) + (1-\alpha)/(\alpha Z)} \end{aligned} \tag{5}$$

We deduce the MLE of $\tilde{c}_s$ which is:

$$\underset{||\tilde{c}_s||=1}{argmax} \sum_{w \in s} f_w(\tilde{c}_s) \propto \sum_{w \in s} \frac{a}{p(w) + a} v_w \tag{6}$$

where $a = \frac{1-\alpha}{\alpha Z}$.

The final c_s estimator is computed by subtracting from the estimator obtained in 6 its projection on the first principal component for the set of estimators.

We note that the SIF weights are inversely proportional to the word frequencies $p(w)$.

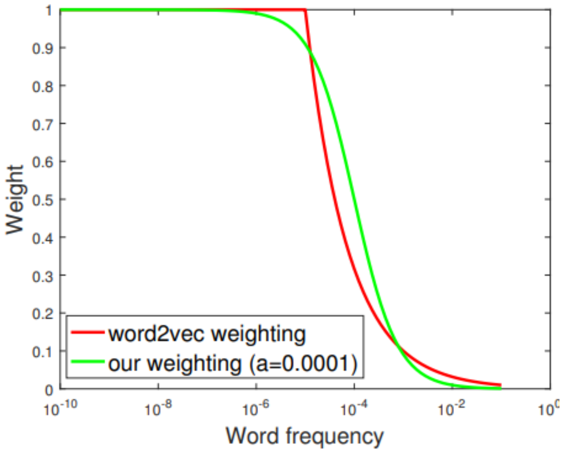

Figure 1: similarity between subsampled word2vec weights and SIF weights. [source [3]]

2 SIF connection to word2vec

In this section we show that we can explain wor2vec embedding (CBOW) with sub-sampling by the SIF weights. The probability model of the CBOW is given by:

$$Pr(w_t/w_{t-1},...,w_{t-5}) \propto exp(<\bar{v}_t, v_w>), \text{ where } \bar{v}_t = \frac{1}{5}\sum_{i=1}^{5} v_{w_{t-i}} \tag{7}$$

For a single word v_w, the loss is written as:

$$g(v_w) = \gamma(<\bar{v}_t, v_w>) + \text{ negative sampling terms} \tag{8}$$

where γ is the logistic function. The sub-sampling method consists in randomly selecting the context words by computed weights. This does not only speed the calculation but also learn more accurate word representations. The sampled update direction is given by:

$$\tilde{\nabla} g(v_w) = \alpha(J_5 v_{w_{t-5}} + J_4 v_{w_{t-4}} + J_3 v_{w_{t-3}} + J_2 v_{w_{t-2}} + J_1 v_{w_{t-1}}) \tag{9}$$

where J_k denotes the Bernoulli random variable with

$$Pr[J_k = 1] = q(w_{t-k}) = min(1, \sqrt{\frac{10^{-5}}{p(w_{t-k})}})$$

By calculating the expectation of the gradient in equation 9, we have:

$$\mathbb{E}[\tilde{\nabla} g(v_w)] = \alpha(q(w_{t-5})v_{w_{t-5}} + q(w_{t-4})v_{w_{t-4}} + q(w_{t-3})v_{w_{t-3}} + q(w_{t-2})v_{w_{t-2}} + q(w_{t-1})v_{w_{t-1}})$$

The above formula corresponds to weighted gradient update. By setting $a = 10^{-4}$ in the SIF weights we can simulate this gradient update (see figure 1). This shows that word2vec with sub-sampling term can be explained using SIF weighting model.

3 Application of the SIF weighting

To verify the success of the weighting method, It was tested on a new task other than the experiments shown in the paper. We used the pretrained word2vec (skip-gram) word embeddings trained on *Wikipedia* using fastext. SIF method was used to weight word embeddings. the α value of the SIF was set to 10^{-3}. We used these embeddings to train a simple encoder that could achieve high accuracy value in classifying fake news which is too comparable to the Universal Sentence Encoder that uses transformer [4]. The accuracy value reached 91%.

The adapted code named "fakenews.ipynb" is available in the link.[1]

4 Critique of the paper

In this part we present some critique review of the presented paper. On this section we refer essentially to the paper [6] of (Alena Sorokina et al. 2019). The critique shows two levels of inconsistencies: on the theoretical approach level and the evaluation level. We discuss then the use of Pearson correlation based on the paper [5].

4.1 Critique of the SIF construction

In the experiments part, in the section 4.1.1 of the paper, authors found out that good results were obtained for:

$$\begin{aligned} 10^{-4} \leq & a \leq 10^{-3} \\ 10^{-4} \leq & \frac{1-\alpha}{\alpha Z} \leq 10^{-3}, \text{ where } Z = \mathbb{E}[Z_c] \end{aligned} \tag{10}$$

In the paper [1], Arora et al. shwed that under isotropic asumption on w's[2]:

$$\mathbb{E}_w[Z_c] = n\mathbb{E}_\xi[exp(\xi^2||c||^2/2)] \tag{11}$$

where ξ is a random variable bounded by a constant.

Combining the inequality 10 and the equation 11, one can prove that for a typical vocabulary of size $n = 10^5$, α is necessarily approximately equal to 1. Implying no dependency w.r.t to the context c. of the probability model in the equation 2.

Another inconsistency shows up in the calculations of MLE estimator (see 1.3). The log-likelihood was linearized around 0 meaning that $\tilde{c}_s \approx 0$. However this approximation contradicts the set on which we maximize the log likelihood.

4.2 Critique of the evaluation

The method used by the paper consists not only in computing the SIF weights of the words but also in subtracting the first component of the PCA. However, at the moment of testing, in any time they performed a subtraction of the first PCA

[1] https://github.com/MohamedISSA98/simple-but-tough-to-beat-examples.git

[2] This asumption is also used in this paper by supposing the uniformly distributed embeddings of words.

components for the baseline representations. which make the comparison inconsistent. In [6], (Alena Sorokina et al.) showed performed the first component removal of both SIF and Avg methods and compared them as well as comparing them before the removal of the PCA component. Results showed diminished advantage of the SIF compared to the Avg.

4.3 Metric critique

To assess textual similarity performance of the model, (Alena Sorokina et al) used the cosine similarity and pearson correlation. However, in [5] (Steffen Eger et al.) show experimentally that this evaluation method can give misleading results. They show in instance that a simple normalization of the data may induce a large difference in the pearson correlation coefficient.

5 Conclusion

SIF weighting method has surely shown better performance leveraging different word embeddings. Although its theoretical setting offers explanations for some word embeddings like word2vec with subsamling, its theory is not too consistent. This method was overtaken on by new models such as the attention model and the transformers which can explain why it was not developed ever since.

References

[1] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

[2] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings, 2019.

[3] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*, 2017.

[4] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[5] Steffen Eger, Andreas Rücklé, and Iryna Gurevych. Pitfalls in the evaluation of sentence embeddings, 2019.

[6] Aidana Karipbayeva, Alena Sorokina, and Zhenisbek Assylbekov. A critique of the smooth inverse frequency sentence embeddings, 2019.