The Document Vectors Using Cosine Similarity Revisited

Zhang Bingyu$^{\triangle}$ **Nikolay Arefyev**$^{\diamond,\triangledown,\triangle}$

$^{\triangle}$National Research University Higher School of Economics / Moscow, Russia
$^{\diamond}$Samsung Research Center Russia / Moscow, Russia
$^{\triangledown}$Lomonosov Moscow State University / Moscow, Russia
bchzhan_1@edu.hse.ru, nick.arefyev@gmail.com

Abstract

The current state-of-the-art test accuracy (97.42%) on the IMDB movie reviews dataset was reported by Thongtan and Phienthrakul (2019) and achieved by the logistic regression classifier trained on the Document Vectors using Cosine Similarity (DV-ngrams-cosine) proposed in their paper and the Bag-of-N-grams (BON) vectors scaled by Naive Bayesian weights. While large pre-trained Transformer-based models have shown SOTA results across many datasets and tasks, the aforementioned model has not been surpassed by them, despite being much simpler and pre-trained on the IMDB dataset only.

In this paper, we describe an error in the evaluation procedure of this model, which was found when we were trying to analyze its excellent performance on the IMDB dataset. We further show that the previously reported test accuracy of 97.42% is invalid and should be corrected to 93.68%. We also analyze the model performance with different amounts of training data (subsets of the IMDB dataset) and compare it to the Transformer-based RoBERTa model. The results show that while RoBERTa has a clear advantage for larger training sets, the DV-ngrams-cosine performs better than RoBERTa when the labelled training set is very small (10 or 20 documents). Finally, we introduce a sub-sampling scheme based on Naive Bayesian weights for the training process of the DV-ngrams-cosine, which leads to faster training and better quality.

1 Introduction

The word2vec algorithm originally published by Mikolov et al. (2013) is among the most famous methods to train vector representations of words. Soon after the emergence of word2vec, a similar method to build vector representations of documents was originally proposed by Le and Mikolov (2014) and further studied by Mesnil et al. (2015). It is known under different names, including Paragraph Vectors, Sentence Vectors, doc2vec, etc. This method jointly learns word embeddings and document embeddings such that a binary classifier can predict if a given word occurs in a particular document given only the corresponding embeddings. More formally, the following objective is minimized:

$$\sum_{d\in D}\sum_{w\in W_d}[-\log\sigma(v_d^T v_w) - \sum_{w'\sim V}\log\sigma(-v_d^T v_{w'})] \quad (1)$$

Here D denotes the set of documents, W_d is the list of words that make up the document d, w' is a word randomly sampled from the full vocabulary V, also known as a negative sample (Goldberg and Levy, 2014). Finally, v_d and v_w are the learnt embeddings of d and w. Intuitively, for each document, an embedding is learnt that has high similarity to the embeddings of those words that occur in this document and low similarity to the embeddings of some random words.

Later Li et al. (2015) switched from single words to n-grams and observed significant improvements. Building on that, Thongtan and Phienthrakul (2019) studied different objective functions. They have found that the cosine similarity outperforms the dot product, which led to a modified model called the Document Vectors using Cosine Similarity (we will call it **DV-ngrams-cosine** for short). The new objective is:

$$\begin{aligned}\sum_{d\in D}\sum_{u\in U_d}&[-\log\sigma(\alpha cos(v_d, v_u))\\ &-\sum_{u'\sim V}\log\sigma(-\alpha cos(v_d, v_{u'}))],\end{aligned} \quad (2)$$

where U_d denotes the set of all n-grams in d, v_u is the embedding of the n-gram u from d, $v_{u'}$ is the embedding of a randomly sampled n-gram, and α is a hyperparameter.

In the same paper, the authors proposed an ensemble consisting of the document embeddings from DV-ngrams-cosine and the Bag-of-N-grams

Proceedings of the Third Workshop on Insights from Negative Results in NLP, pages 129 - 133
May 26, 2022 ©2022 Association for Computational Linguistics

vectors scaled by Naive Bayesian weights (**NB-weighted BON** for short). They concatenated these two representations and trained the logistic regression classifier on top. The ensemble was reported to have very high test accuracy (97.42%) on the IMDB movie reviews dataset (Maas et al. (2011)). To the best of our knowledge, this accuracy remains the SOTA result on IMDB. Even large Transformer-based models pre-trained on a huge amount of texts, both in-domain and out-of-domain, have shown lower accuracy on this dataset (Yang et al., 2019; Suchin et al., 2020; Arefyev et al., 2021).

This extraordinary performance of such a simple model motivated us to thoroughly study the model and its implementation trying to understand the reasons behind its success. Unfortunately, during this study, we found a bug in the implementation of the evaluation procedure of the ensemble, which had made the estimation of the accuracy incorrect.

In our paper, we re-evaluate the ensemble as well as its individual components. We show that the originally reported test accuracy of the ensemble (97.42%) is incorrect and shall be corrected to 93.68%, which is only 0.55% higher than the accuracy on pure DV-ngrams-cosine embeddings.

Additionally, we analyze how the amount of training data affects the performance of the ensemble, as well as its individual components, and also the Transformer-based RoBERTa model (Liu et al., 2020), which has recently shown SOTA or near-SOTA results over a variety of tasks and datasets. Surprisingly, we have observed that DV-ngrams-cosine outperforms RoBERTa when the number of labelled training examples is small (10 or 20). We also ensemble RoBERTa with DV-ngrams-cosine, but only have achieved a marginal improvement. Finally, we propose a modification for the training process of DV-ngrams-cosine that results in faster training and better accuracy. The code reproducing our experiments is publicly available [1].

2 Re-evaluation of the ensemble

In the aforementioned ensemble proposed by Thongtan and Phienthrakul (2019), the NB-weighted BON and the DV-ngrams-cosine are concatenated and fed into the logistic regression classifier. However, we have found that in the original implementation the two vectors concatenated to obtain a single training or test example usually correspond to two different documents of the same class (see details in Appendix A). Specifically, the DV-ngrams-cosine vectors and the BON vectors are built from two different files having different orders of examples. As a result, after the concatenation, each input to the logistic regression corresponds to a combination of two examples. Due to the special structure of the files, those examples are guaranteed to belong to the same class and the same subset. For instance, a positive example from the test set is concatenated with another positive example from the test set.

In Appendix B.3 we provide an analysis that shows the reasons of high performance of this concatenation of two representations. From this analysis it follows that most examples from IMDB are correctly classified with high confidence (a large logit) using any of two representations, i.e. they are easy examples. Less than 10% of examples are classified incorrectly by each representation (hard examples), but they often obtain low confidence (a logit near zero). Hard examples are more often combined with easy examples just because of their dominance. In these cases, the logit from the easy example often outweigh the logit from the hard one resulting in the correct final prediction.

Thus, in both the training and the test sets, hard examples are often combined with simpler examples, making the classification task easier. In this process, the knowledge of the true labels is implicitly exploited to combine the examples this way, in both training and testing. This leads to an incorrect estimation of the classification accuracy for future examples.

After fixing this issue, we have observed that the combination of different representations of the same document leads to the test accuracy of 93.68% instead of 97.42% originally reported. Compared to the pure DV-ngrams-cosine embeddings, the ensemble improves the test accuracy by 0.55%, not 4.29% reported previously. This improvement also better agrees with the improvements of less than 1% observed by Li et al. (2015) for similar ensembles with the predecessor model DV-ngram. As a sanity check, Appendix B additionally reports the accuracy for different schemes of combining the two representations, showing that higher accuracy can be achieved only by those schemes that exploit the knowledge of the test labels.

[1] https://github.com/Bgzh/dv_cosine_revisited

3 Further analysis of performance

In his section we further analyze the performance of the ensemble described above, comparing it to its individual components as well as to the recently introduced Transformer-based RoBERTa model (Liu et al., 2020). We study the performance of these models depending on the number of labelled examples in the training set.

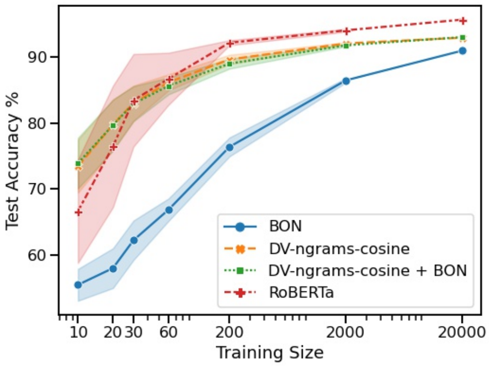

Figure 1: The performance of different models on training sets of different sizes. The mean values and standard deviations were calculated over 10 random subsets for RoBERTa and 30 random subsets for other models for each training set size. BON in the legend implies NB-weighted BON.

For a more fair comparison, the most important hyperparameters of each model were tuned on the validation set, employing the train/validation/test split of the IMDB dataset provided by (Suchin et al., 2020). Subsets of different sizes from 10 to 20000 examples were randomly sampled from the training set. The logistic regression classifier was trained on these subsets using the DV-ngram-cosine embeddings, the NB-weighted BON vectors, or their concatenation as its input representation.

We tuned the L2-regularization strength C of the classifier individually for each subset of the training set. Additionally, we multiplied the DV-ngram-cosine embeddings before concatenating them to the BON vectors in order to balance the magnitudes of the two representations, which may help the classifier to benefit from both representations. The scaling factor was also selected on the validation set.

The pre-trained RoBERTa base model[2] was fine-tuned on a part (10 out of 30) of the same subsets of the training set, using the validation set for early stopping. We used a batch size of 32, with a maximum learning rate of 1e-5, recommended by fairseq[3].

As shown in Fig. 1, the fine-tuned RoBERTa model usually achieves higher test accuracy. But when the number of labelled training examples is very small (10 or 20), the logistic regression on the DV-ngrams-cosine embeddings shows higher mean test accuracy and lower standard deviation. This result corroborated the notion that small models can be a better choice when the data are scarce.

On the other hand, logistic regression on the BON vectors performs significantly worse than all other models across all training set sizes. Finally, we don't observe any significant improvements from the ensembling when the training set size is less than 20k, as the difference is within one standard deviation.

It is important to notice that the DV-ngrams-cosine embeddings were pre-trained on the in-domain examples from the whole IMDB dataset, while RoBERTa was pre-trained on a huge but general-domain corpus. It is likely that the domain adaptation techniques (Suchin et al., 2020) will help RoBERTa when the number of labelled examples is small. However, for our study, we decided to compare the most standard approaches to training the corresponding models.

4 NB Sub-Sampling

In this section, we improve the training procedure of DV-ngrams-cosine by applying a sub-sampling procedure based on the Naive Bayesian weights of ngrams (**NB Sub-Sampling**) in order to make the model focus more on sentiment-related ngrams while building the document embeddings.

Inspired by the previous works (Wang and Manning (2012), Arefyev et al. (2021)), we trained a multinomial Naive Bayesian Classifier and exploited its weights to calculate the importance of each ngram f_i for the final classification task:

$$h_i = |\log p(f_i|y=1) - \log p(f_i|y=0)| \quad (3)$$

In each epoch we put an ngram into training with the probability

$$p(f_i) = min(\exp(h_i/n_a)/n_b, 1), \quad (4)$$

[2] https://pytorch.org/hub/huggingface_pytorch-transformers/

[3] https://github.com/pytorch/fairseq/blob/main/examples/roberta/README.custom_classification.md

Model	Test Accuracy %
Models trained on the original training set of IMDB (25K)	
NB-weighted BON	91.29
DV-ngrams-cosine	93.13
DV-ngrams-cosine + NB-weighted BON (Thongtan and Phienthrakul, 2019)	#97.42
DV-ngrams-cosine + NB-weighted BON (re-evaluated)	93.68
Models trained using the train/dev split from (Suchin et al., 2020) (20K/5K)	
DV-ngrams-cosine with NB sub-sampling	93.36
RoBERTa	95.79
DV-ngrams-cosine + RoBERTa	95.92
DV-ngrams-cosine with NB sub-sampling + RoBERTa	95.94

Table 1: Test results on the IMDB dataset. # indicates incorrect previously reported results.

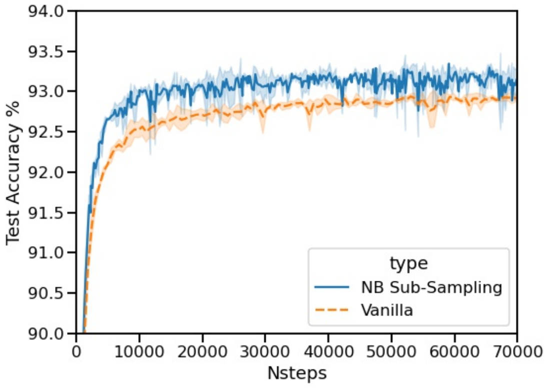

Figure 2: Training process with and without NB sub-sampling. The test accuracy of the logistic regression built on top of the document vectors is plotted. The mean values and standard deviations were calculated over 3 runs for each type.

where n_a and n_b are the hyperparameters. The choices are purely empirical. We tried different combinations of n_a and n_b and found 2 and 3 (respectively) to be the best in them.

The comparison of the training process with and without NB sub-sampling is shown in Fig. 2 (refer to Appendix C for details of the experiments and the accuracy on the validation set).

The runs with NB sub-sampling progress faster and show a distinct advantage after 2500 steps. After 30k steps, the runs with NB sub-sampling stagnated and kept fluctuating in a small region; the vanilla runs stagnated after 50k steps, in a lower area. It is also worth noticing that although the labels of the training set are used during pre-training for sub-sampling, we did not observe any significant overfitting due to that. Neither the validation score nor the test score showed a tendency to decay long after reaching the plateau, indicating that this sub-sampling scheme can be used as an add-on to the original model, boosting its performance while not creating additional overfitting trouble.

5 Ensemble DV-ngrams-cosine and RoBERTa

The ensemble proposed in (Thongtan and Phienthrakul (2019)) and described in Section 2 combines two different representations of documents, which are the DV-ngrams-cosine embeddings and the NB-weighted BON vectors. However, we have observed in Section 3 that the BON vectors are quite weak on their own, while RoBERTa outperforms all other models unless the number of examples is very small. Thus, it is interesting if DV-ngram-cosine can help RoBERTa. In this section, we combine the DV-ngrams-cosine (with or without NB sub-sampling) with the output of the last hidden layer of RoBERTa, and test on the IMDB dataset. Again, the train/validation/test splits by Suchin et al. (2020) were used. A scaling factor on the DV-ngrams-cosine and the hyperparameter C in the logistic regression were tuned on the validation set.

The results are shown in Table 1. Although RoBERTa is a much stronger model than DV-ngram-cosine, combining them has shown a small improvement of 0.13-0.15%.

6 Conclusion

The ensemble featuring the DV-ngrams-cosine reported by Thongtan and Phienthrakul (2019) was re-evaluated. The test accuracy of this ensemble on the IMDB dataset was corrected from 97.42% to 93.68%. The DV-ngrams-cosine embeddings with the logistic regression on top were compared with RoBERTa using different amounts of training data.

In this comparison, the DV-ngrams-cosine has surprisingly outperformed RoBERTa for a small number of training examples (10 or 20 documents). A sub-sampling scheme based on the Naive Bayesian weights was introduced to the training process of the DV-ngrams-cosine, resulting in faster training and better quality.

Acknowledgements

We are grateful to our anonymous reviewers. This research was partially supported by the Basic Research Program at the HSE University.

References

Nikolay Arefyev, Dmitry Kharchev, and Artem Shelmanov. 2021. Nb-mlm - efficient domain adaptation of masked language models for sentiment analysis. *EMNLP*, pages 9114–9124.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *CoRR*.

V. Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *ICML*, pages 1188–1196.

Bofang Li, Tao Liu, Xiaoyong Du, Deyuan Zhang, and Zhe Zhao. 2015. Learning document embeddings by predicting n-grams for sentiment classification of long movie reviews. *CoRR*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

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 *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Grégoire Mesnil, Tomas Mikolov, Marc'Aurelio Ranzato, and Yoshua Bengio. 2015. Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. *international conference on learning representations*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*.

Gururangan Suchin, Marasović Ana, Swayamdipta Swabha, Lo Kyle, Beltagy Iz, Downey Doug, and Noah Smith A. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *ACL*, pages 8342–8360.

Tan Thongtan and Tanasanee Phienthrakul. 2019. Sentiment classification using document embeddings trained with cosine similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 407–414, Florence, Italy. Association for Computational Linguistics.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, G. Jaime Carbonell, Ruslan Salakhutdinov, and V. Quoc Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 32 (NIPS 2019)*, pages 5754–5764.