Abstract

Producing the embedding of a sentence in an unsupervised way is valuable to natural language matching and retrieval problems in practice. In this work, we conduct a thorough examination of pretrained model based unsupervised sentence embeddings. We study on four pretrained models and conduct massive experiments on seven datasets regarding sentence semantics. We have three main findings. First, averaging all tokens is better than only using $[CLS]$ vector. Second, combining both top and bottom layers is better than only using top layers. Lastly, an easy whitening-based vector normalization strategy with less than 10 lines of code consistently boosts the performance.

1 Introduction

Pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019) perform well on learning sentence semantics when fine-tuned with supervised data (Reimers and Gurevych, 2019; Thakur et al., 2020). However, in practice, especially when a large amount of supervised data is unavailable, an approach that provides sentence embeddings in an unsupervised way is of great value in scenarios like sentence matching and retrieval. We explore whether using the hidden vectors of other tokens is beneficial. Second, some works suggest producing sentence embedding from the last layer or the combination of the last two layers (Reimers and Gurevych, 2019; Li et al., 2020). We seek to figure out whether there exists a better way of layer combination. Third, recent attempts transform sentence embeddings to a different distribution with sophisticated networks (Li et al., 2020) to address the problem of non-smooth anisotropic distribution. Instead, we aim to explore whether a simple linear transformation is sufficient.

To answer these questions, we conduct thorough experiments upon 4 different PLMs and evaluate on 7 datasets regarding semantic textual similarity. We find that, first, to average the token representations consistently yields better sentence representations than using the representation of the $[CLS]$ token. Second, combining the embeddings of the bottom layer and the top layer performs than using top two layers. Third, normalizing sentence embeddings with whitening, an easy linear matrix transformation algorithm with less than 10 lines of code, consistently brings improvements.

2 Transformer-based PLMs

Multi-layer Transformer architecture (Vaswani et al., 2017) has been widely used in pre-trained language models (e.g. Devlin et al., 2019; Liu et al., 2019) to encode sentences. Given an input sequence $S = \{s_1, s_2, \ldots, s_n\}$, a transformer-based PLM produces a set of hidden representations $H^{(0)}, H^{(1)}, \ldots, H^{(L)}$, where $H^{(l)} = [h_1^{(l)}, h_2^{(l)}, \ldots, h_n^{(l)}]$ are the per-token embeddings of $S$ in the $l$-th encoder layer and $H^{(0)}$ corresponds to the non-contextual word(piece) embeddings.

In this paper, we investigate PLMs-based unsupervised sentence embeddings from three aspects. First, a standard way of obtaining sentence embedding is to pick the vector of $[CLS]$ token. We
Whitening is a linear transformation that transforms word embedding mapping (Artetxe et al., 2018) with two layers, we obtain sentence embeddings. Taking the last layer of token representations as vectors of all tokens in the sentence, including the vector of \([3.2] \text{Layer Combination}

Most works only take the last layer to derive sentence embeddings, while rarely explore which layer of semantic representations can help to derive a better sentence embedding. Here we explore how to best combine layers of embeddings to obtain sentence embeddings. Specifically, we can first compute the vector representation of each layer following Section 3.1. Then we perform layer combinations as \(s = \sum_i s^i\) to acquire sentence embedding. For example, for the combination of L1+L12 with two layers, we obtain sentence embeddings by adding up the vector representation of layer one and layer twelve, i.e., \(s = \frac{1}{2} (s^1 + s^{12})\).

3.3 Whitening

Whitening is a linear transformation that transforms a vector of random variables with a known covariance matrix into a new vector whose covariance is an identity matrix, and has been verified effective to improve the text representations in bilingual word embedding mapping (Artetxe et al., 2018) and image retrieval (Jégou and Chum, 2012).

In our work, we explore to address the problem of non-smooth anisotropic distribution (Li et al., 2020) by a simple linear transformation method called whitening. Specifically, given a set of embeddings of \(N\) sentences \(\mathbf{E} = \{s_1, \ldots, s_N\} \in \mathbb{R}^{N \times d}\), where \(d\) is the dimension of the embedding, we transform \(\mathbf{E}\) linearly as in Eq. 1 such that \(\hat{\mathbf{E}} \in \mathbb{R}^{N \times d}\) is the whitened sentence embeddings,

\[
\hat{\mathbf{E}} = (\mathbf{E} - \mathbf{m}) \mathbf{U} \mathbf{D}^{-\frac{1}{2}},
\]

where \(\mathbf{m} \in \mathbb{R}^d\) is the mean vector of \(\mathbf{E}\), \(D\) is a diagonal matrix with the eigenvalues of the covariance matrix \(\text{Cov}(\mathbf{E}) = (\mathbf{E} - \mathbf{m})^T (\mathbf{E} - \mathbf{m}) \in \mathbb{R}^{d \times d}\) and \(U\) is the corresponding orthogonal matrix of eigenvectors, satisfying \(\text{Cov}(\mathbf{E}) = \mathbf{U} \mathbf{D} \mathbf{U}^T\).

4 Experiment

We evaluate sentence embeddings on the task of unsupervised semantic textual similarity. We show experimental results and report the best way to derive unsupervised sentence embedding from PLMs.

4.1 Experiment Settings

Task and Datasets The task of unsupervised semantic textual similarity (STS) aims to predict the similarity of two sentences without direct supervision. We experiment on seven STS datasets, namely the STS-Benchmark (STS-B) (Cer et al., 2017), the SICK-Relatedness (Marelli et al., 2014), and the STS tasks 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016). These datasets consist of sentence pairs with labeled semantic similarity scores ranging from 0 to 5.

Evaluation Procedure Following the procedures in previous works like SBERT (Reimers and Gurevych, 2019), we first derive sentence embeddings for each sentence pair and compute the cosine similarity score of the embeddings as the predicted similarity. Then we calculate the Spearman’s rank correlation coefficient between the predicted similarity and gold standard similarity scores as the evaluation metric. We average the Spearman’s coefficients among the seven datasets as the final correlation score.

Baseline Methods We compare our methods with five representative unsupervised sentence embedding models, including average GloVe embedding (Pennington et al., 2014), SIF (Arora et al., 2017), IS-BERT (Zhang et al., 2020) and BERT-flow (Li et al., 2020), SBERT-WK with BERT (Wang and Kuo, 2020).

4.2 Overall Results

Table 1 shows the overall performance of sentence embeddings with different models and settings. We can observe that:
which indicates that single layer combination is capable of fusing the semantic information in different layers and thus yields better performance. Therefore, we suggest summing up the last layer and layer one to perform layer combination and induce better sentence embeddings.

Adding up the token representations in layer one and the last layer to form the sentence embeddings performs better than separately using only one layer, regardless of the selection of the PLM. Since PLMs capture a rich hierarchy of linguistic information in different layers (Tenney et al., 2019; Jawahar et al., 2019), layer combination is capable of fusing the semantic information in different layers and thus yields better performance. Therefore, we suggest summing up the last layer and layer one to perform layer combination and induce better sentence embeddings.

Introducing the whitening strategy produces consistent improvement of sentence embeddings on STS tasks. This result indicates the effectiveness of the whitening strategy in deriving sentence embeddings. Among the four PLMs, LaBSE achieves the best STS performance while obtaining the least performance enhancement after incorporating whitening strategy. We attribute it to the good intrinsic representation ability because LaBSE is pre-trained by a translation ranking task which improves the sentence embedding quality.

Table 1: Spearman’s rank correlation coefficient ($\rho \times 100$) between similarity scores assigned by sentence embeddings and humans. $token=AVG$ or $token=CLS$ denote using the average vectors of all tokens or only the $[CLS]$ token. $L1$ or $L12$ ($L6$) means using the hidden vectors of layer one or the last layer. Since DistilBERT only contains six layers of transformers, we use $L6$ as the last layer. T and F denote applying whitening (T) or not (F).

| Models                                      | STSB  | SICK  | STS-12 | STS-13 | STS-14 | STS-15 | STS-16 | Avg.   |
|---------------------------------------------|-------|-------|--------|--------|--------|--------|--------|--------|
| **Baselines**                               |       |       |        |        |        |        |        |        |
| Avg. GloVe (Reimers and Gurevych, 2019)     | 58.02 | 53.76 | 55.14  | 70.66  | 59.73  | 68.25  | 63.66  | 61.32  |
| SIF (GloVe+WR) (Arora et al., 2017)        | -     | -     | -      | -      | -      | -      | -      | -      |
| IS-BERT-NLI (Zhang et al., 2020)           | 69.21 | 64.25 | 56.77  | 69.24  | 61.21  | 75.23  | 70.16  | 66.58  |
| BERT-flow (NL) (Li et al., 2020)           | 58.36 | 65.44 | 59.54  | 64.69  | 64.66  | 72.92  | 71.84  | 65.38  |
| SBERT WK (BERT) (Wang and Kuo, 2020)       | 16.07 | 41.54 | 26.66  | 14.74  | 24.32  | 28.84  | 34.37  | 26.65  |
| **WhiteningBERT (PLM=BERT-base)**          |       |       |        |        |        |        |        |        |
| $token=CLS$, $layer=L12$, $whitening=F$     | 20.29 | 42.42 | 32.50  | 23.99  | 28.50  | 35.51  | 51.08  | 33.47  |
| $token=AVG$, $layer=L12$, $whitening=F$     | 47.29 | 58.22 | 50.08  | 52.91  | 54.91  | 63.37  | 64.94  | 55.96  |
| $token=AVG$, $layer=L1$, $whitening=F$      | 58.15 | 61.78 | 58.71  | 58.21  | 62.51  | 68.86  | 67.38  | 62.23  |
| $token=AVG$, $layer=L1+L12$, $whitening=F$ | 59.05 | 63.75 | 57.72  | 58.38  | 61.97  | 70.28  | 69.63  | 62.97  |
| $token=AVG$, $layer=L1+L12$, $whitening=T$ | 68.68 | 60.28 | 61.94  | 68.47  | 67.31  | 74.82  | 72.82  | 67.76  |
| **WhiteningBERT (PLM=RoBERTa-base)**       |       |       |        |        |        |        |        |        |
| $token=CLS$, $layer=L12$, $whitening=F$     | 38.80 | 61.89 | 45.38  | 36.25  | 47.99  | 53.94  | 59.48  | 49.10  |
| $token=AVG$, $layer=L12$, $whitening=F$     | 55.43 | 62.03 | 53.80  | 46.55  | 56.61  | 64.97  | 63.61  | 57.57  |
| $token=AVG$, $layer=L1$, $whitening=F$      | 51.85 | 57.87 | 56.70  | 48.03  | 57.08  | 62.83  | 57.64  | 56.00  |
| $token=AVG$, $layer=L1+L12$, $whitening=F$ | 57.54 | 60.75 | 58.56  | 50.37  | 59.62  | 66.64  | 63.21  | 59.53  |
| $token=AVG$, $layer=L1+L12$, $whitening=T$ | 69.43 | 59.56 | 62.66  | 62.94  | 68.44  | 74.89  | 72.94  | 67.72  |
| **WhiteningBERT (PLM=DistilBERT)**         |       |       |        |        |        |        |        |        |
| $token=CLS$, $layer=L6$, $whitening=F$      | 30.96 | 47.73 | 40.91  | 31.30  | 39.49  | 40.64  | 57.96  | 41.29  |
| $token=AVG$, $layer=L6$, $whitening=F$      | 57.17 | 63.53 | 56.16  | 59.83  | 60.42  | 67.81  | 69.01  | 61.99  |
| $token=AVG$, $layer=L1$, $whitening=F$      | 55.35 | 61.34 | 57.57  | 53.79  | 60.55  | 67.06  | 63.60  | 59.89  |
| $token=AVG$, $layer=L1+L6$, $whitening=F$  | 61.45 | 63.84 | 59.67  | 59.50  | 63.54  | 70.95  | 69.90  | 64.12  |
| $token=AVG$, $layer=L1+L6$, $whitening=T$  | 70.37 | 58.31 | 62.09  | 68.78  | 68.99  | 75.06  | 74.52  | 68.30  |
| **WhiteningBERT (PLM=LaBSE)**              |       |       |        |        |        |        |        |        |
| $token=CLS$, $layer=L12$, $whitening=F$     | 67.18 | 69.43 | 66.99  | 61.26  | 68.36  | 77.13  | 73.10  | 69.06  |
| $token=AVG$, $layer=L12$, $whitening=F$     | 71.02 | 68.36 | 67.81  | 63.94  | 70.56  | 77.93  | 75.07  | 70.67  |
| $token=AVG$, $layer=L1$, $whitening=F$      | 53.70 | 55.25 | 54.81  | 44.62  | 56.97  | 60.30  | 54.57  | 54.32  |
| $token=AVG$, $layer=L1+L12$, $whitening=F$ | 72.56 | 68.36 | 68.30  | 65.75  | 71.41  | 78.90  | 75.68  | 71.56  |
| $token=AVG$, $layer=L1+L12$, $whitening=T$ | 73.32 | 63.27 | 68.45  | 71.11  | 71.66  | 79.30  | 74.87  | 71.71  |
Figure 1: Performance of sentence embeddings of two layers of combinations. X-axis and Y-axis denote the layer index. Each cell is the average correlation score of seven STS tasks of two specific layer combinations. The redder the cell is, the better performance the corresponding sentence embeddings achieve.

Figure 2: Maximum correlation scores of sentence embeddings from BERT-base with different numbers of combining layers. Combining three layers performs best than other layer numbers. Especially the best combination is L1+L2+L12.

4.3 Analysis of Layer Combination

To further investigate the effects of layer combination, we add up the token representations of different layers to induce sentence embeddings.

First, we explore whether adding up layer one and the last layer is consistently better than other combinations of two layers. Figure 1 shows the performance of all two-layer combinations. We find that adding up the last layer and layer one do not necessarily achieve the best performance among all PLMs, but could be a satisfying choice for simplicity.

Second, we explore the effects of the number of layers to induce sentence embeddings. We evaluate on BERT-base and figure 2 shows the maximum correlation score of each group of layer combinations. By increasing the number of layers, the maximum correlation score increases first but then drops. The best performance appears when the number of layers is three (L1+L2+L12). This indicates that combining three layers is sufficient to yield good sentence representations and we do not need to incorporating more layers which is not only complex but also poorly performed.

5 Related works

Unsupervised sentence embeddings are mainly composed with pre-trained (contextual) word embeddings (Pennington et al., 2014; Devlin et al., 2019). Recent attempts can be divided into two categories, according to whether the pre-trained embeddings are further trained or not. For the former, some works leverage unlabelled natural language inference datasets to train a sentence encoder without direct supervision (Li et al., 2020; Zhang et al., 2020; Mu and Viswanath, 2018). For the latter, some works propose weighted average word embeddings based on word features (Arora et al., 2017; Ethayarajh, 2018; Yang et al., 2019; Wang and Kuo, 2020). However, these approaches need further training or additional features, which limits the direct applications of sentence embeddings in real-world scenarios. Finally, we note that concurrent to this work, Su et al. (2021) also explored whitening sentence embedding, released to arXiv.
one week before our paper.

6 Conclusion

In this paper, we explore different ways and find a simple and effective way to produce sentence embedding upon various PLMs. Through exhaustive experiments, we make three empirical conclusions here. First, averaging all token representations consistently induces better sentence representations than using the $[CLS]$ token embedding. Second, combining the embeddings of the bottom layer and the top layer outperforms that using the top two layers. Third, normalizing sentence embeddings with a whitening algorithm consistently boosts the performance.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Matthew Cer, Mona T. Diab, A. Gonzalez-Agirre, Weiwei Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, German Rigau, L. Uria, and J. Wiebe. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In SemEval@NAACL-HLT.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Matthew Cer, Mona T. Diab, A. Gonzalez-Agirre, Weiwei Guo, R. Mihalcea, German Rigau, and J. Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In SemEval@COLING.

Eneko Agirre, Carmen Banea, Daniel Matthew Cer, Mona T. Diab, A. Gonzalez-Agirre, R. Mihalcea, German Rigau, and J. Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In SemEval@NAACL-HLT.

Eneko Agirre, Daniel Matthew Cer, Mona T. Diab, and A. Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In SemEval@NAACL-HLT.

Eneko Agirre, Daniel Matthew Cer, Mona T. Diab, A. Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. In *SEM@NAACL-HLT.

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

M. Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In AAAI.

T. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, T. Henighan, R. Child, A. Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Rafford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. ArXiv, abs/2005.14165.

Daniel Matthew Cer, Mona T. Diab, Eneko Agirre, I. Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In SemEval@ACL.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. ArXiv, abs/2004.07180.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT.

Kawin Ethayarajh. 2018. Unsupervised random walk sentence embeddings: A strong but simple baseline. In Rep4NLP@ACL.

Fangxiaoyu Feng, Yin-Fei Yang, Daniel Matthew Cer, N. Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. ArXiv, abs/2007.01852.

Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and B. Dolan. 2020. Dialogue response ranking training with large-scale human feedback data. In EMNLP.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. ArXiv, abs/2006.03654.

Forrest N. Iandola, Albert Eaton Shaw, R. Krishna, and K. Keutzer. 2020. Squeezebert: What can computer vision teach nlp about efficient neural networks? ArXiv, abs/2006.11316.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In ACL.

H. Jégou and O. Chum. 2012. Negative evidences and co-occurrences in image retrieval: The benefit of pca and whitening. In ECCV.

Mandar Joshi, Danqi Chen, Y. Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics, 8:64–77.
A Appendix

A.1 More Results of WhiteningBERT

To further illustrate the effectiveness of the whitening algorithm in induce sentence embeddings for STS tasks, we experiment with more PLMs and report their performance with and without incorporating the whitening algorithm. From the results exhibited in Table 2, we find that no matter which PLM we use, the average performance on 7 STS tasks improves after incorporating the whitening strategy. This result again verifies the effectiveness of whitening in producing sentence embeddings.
def whitening_torch(embodings):
    mu_torch = torch.mean(embodings, dim=0, keepdim=True)
    cov_torch = torch.mm(embodings - mu_torch, embodings - mu_torch)
    u, s_torch = torch.svd(cov_torch)
    W_torch = torch.mm(u_torch, torch.diag(torch.sqrt(s_torch)))
    return embodings

Figure 3: Pytorch code for whitening strategy.