Connecting Supervised and Unsupervised Sentence Embeddings

Gil Levi
Tel Aviv University
gil.levi100@gmail.com

Abstract

Representing sentences as numerical vectors while capturing their semantic context is an important and useful intermediate step in natural language processing. Representations that are both general and discriminative can serve as a tool for tackling various NLP tasks.

While common sentence representation methods are unsupervised in nature, recently, an approach for learning universal sentence representation in a supervised setting was presented in (Conneau et al., 2017). We argue that although promising results were obtained, an improvement can be reached by adding various unsupervised constraints that are motivated by auto-encoders and by language models. We show that by adding such constraints, superior sentence embeddings can be achieved. We compare our method with the original implementation and show improvements in several tasks.

1 Introduction

Word embeddings are considered one of the key building blocks in natural language processing and are widely used for various applications (Mikolov et al., 2013; Pennington et al., 2014). While word representations has been successfully used, representing the more complicated and nuanced nature of the next element in the hierarchy - a full sentence - is still considered a challenge. Once trained, universal sentence representations can be used as an out-of-the-box tool for solving various NLP and computer vision problems. Even though their importance is unquestionable, it seems that current results are still far from satisfactory.

More concretely, given a set of sentences $\{s_i\}_{i=1}^n$, sentence embedding methods are designed to map them to some feature space $\mathcal{F}$ along with a distance metric $\mathcal{M}$ such that given two sentences s_i and s_j that have similar semantic meaning, their distance $\mathcal{M}(s_i, s_j)$ would be small. The challenge is learning a mapping $\mathbf{T} : \{s_i\}_{i=1}^n \rightarrow \mathcal{F}$ that manages to capture the semantics of each s_i. While sentence embedding are not always used in similarity probing, we find this formulation useful as the similarity assumption is implicitly made when training classifiers on top of the embeddings in downstream tasks.

Sentences embedding methods were mostly trained in an unsupervised setting. In (Le and Mikolov, 2014) the ParagraphVector model was proposed which is trained to predict words in the document. SkipThought (Kiros et al., 2015) vectors rely on the continuity of text to train an encoder-decoder model that tries to reconstruct the surrounding sentences of a given passage. In Sequential Denoising Autoencoders (SDAE) (Hill et al., 2016) high-dimensional input data is corrupted according to some noise function, and the model is trained to recover the original data from the corrupted version. FastSent (Hill et al., 2016) learns to predicts a Bag-Of-Word (BOW) representation of adjacent sentences given a BOW representation of some sentence. In (Klein et al., 2015) a Hybrid Gaussian Laplacian density function is fitted to the sentence to derive Fisher Vectors.

While previous methods train sentence embeddings in an unsupervised manner, a recent work (Conneau et al., 2017) argued that better representations can be achieved via supervised training on a general sentence inference dataset (Bowman et al., 2015). To this end, the authors use the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015) to train different

Proceedings of the 3rd Workshop on Representation Learning for NLP, pages 79–83
Melbourne, Australia, July 20, 2018. ©2018 Association for Computational Linguistics

Method	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	SICK-R	SICK-E	STS14
FastSent	70.8	78.4	88.7	80.6	-	76.8	72.2/80.3	-	-	.63/.64
SkipThought	76.5	80.1	**93.6**	87.1	82.0	**92.2**	73.0/82.0	0.858	82.3	.29/.35
BiLSTM	79.9	84.6	92.1	89.8	83.3	88.7	75.1/82.3	0.885	86.3	**.68/.65**
AE Reg	79.0	84.4	91.8	90.0	82.4	88.8	75.0/82.4	**0.888**	86.8	.66/.65
LM Reg	79.1	**85.3**	92.2	**90.2**	83.6	87.6	75.7/**82.8**	**0.888**	86.3	.66/.65
Combined	**80.04**	84.56	91.96	90.19	**84.07**	87.8	74.84/82.34	0.888	86.44	.67/.65
Bi-AE Reg	**79.9**	84.1	92.1	**90.2**	83.8	89	**75.9**/82.6	**0.888**	**87.7**	.66/.65
Bi-LM Reg	79.1	84.6	91.2	90.0	82.6	89.4	74.4/81.8	**0.888**	86.4	.66/.64

Table 1: **Sentence embedding results.** BiLSTM refers to the original BiLSTM followed by Max-Pooling implementation of (Conneau et al., 2017) which is the baseline for our work. AE Reg and LM Reg refers to the Auto-Encoder and Language-Model regularization terms described in 2.1 and Combined refers to optimizing with both terms. Bi-AE Reg and Bi-LM Reg refers to the bi-directional Auto-Encoder and bi-directional Language-Model regularization terms described in 2.2. As evident from the results, adding simple unsupervised regularization terms improves the results of the model on almost all the evaluated tasks.

sentence embedding methods and compare them on various benchmarks. The SNLI dataset is composed of 570K pairs of sentences with a label depicting the relationship between them, which can be either 'neutral', 'contradiction' or 'entailment'. The authors show that by leveraging the dataset, state-of-the-art representations can be obtained which are universal and general enough for solving various NLP tasks.

A different, unsupervised, task in NLP is estimating the probability of word sequences. A family of algorithms for this task titled *word language models* seek to model the problem as estimating the probability of a word, given the previous words in the text. In (Bengio et al., 2003) neural networks were employed and (Mikolov et al., 2010) was among the first methods to use recurrent neural networks (RNN) for modeling the problem, where the probability of the a word is estimated based on the previous words fed to the RNN. A variant of RNN - Long Short Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) - were used in (Sundermeyer et al., 2012). Following that, (Zaremba et al., 2014) proposed a dropout augmented LSTM.

We note that there exists a connection between those two problems and try to model it more explicitly. Recently, the incorporation of the hidden states of neural language models in downstream supervised-learning models have been shown to improve the results of the latter (e.g. ElMo - Peters et al. (2018), CoVe - McCann et al. (2017) Peters et al. (2017), Salant and Berant (2017)) – in this work we jointly train the unsupervised and supervised tasks. To this end, we incorporate unsupervised regularization terms motivated by language modeling and auto-encoders in the training framework proposed by (Conneau et al., 2017). We test our proposed model on a set of NLP tasks and show improved results over the baseline framework of (Conneau et al., 2017).

2 Method

Our approach builds upon the previous work of (Conneau et al., 2017). Specifically, we use their BiLSTM model with max pooling. More concretely, given a sequence of **T** words, $\{w_t\}_{t=1,\dots,T}$ with given word embedding (Mikolov et al., 2013; Pennington et al., 2014) $\{v_t\}_{t=1,\dots,T}$,a bidirectional LSTM computes a set of **T** vectors $\{h_t\}_{t=1,\dots,T}$ where each h_t is the concatenation of a forward LSTM and a backward LSTM that read the sentences in two opposite directions. We denote $\{\overrightarrow{h_t}\}$ and $\{\overleftarrow{h_t}\}$ as the hidden states of the left and right LSTM's respectively, where $t = 1, \dots, T$. The final sentence representation is obtained by taking the maximal value of each dimension of the $\{h_t\}$ hidden units (i.e.: max pooling). The original model of (Conneau et al., 2017) was trained on the SNLI dataset in a supervised fashion - given pairs of sentences s_1 and s_2, denote their representation by $\bar{s_1}$ and $\bar{s_2}$. During training, the concatenation of $\bar{s_1}$, $\bar{s_2}$, $|\bar{s_1} - \bar{s_2}|$ and $\bar{s_1} * \bar{s_2}$ is fed to a three layer fully connected network followed by a softmax classifier.

2.1 Regularization terms

We note that by training on SNLI, the model might overfit and would not be general enough to provide universal sentence embedding. We devise several regularization criteria that incentivize the hidden states to maintain more information about the input sequence.

Specifically, denote the dimension of the word embedding by d and the dimension of the hidden state by l. We add a linear transformation layer $L_{l\times d} : H \rightarrow W$ on top of the BiLSTM to transform the hidden states back to the dimension of word embeddings and denote its output by $\{w'_t\}_{t=1,\dots,T}$. Recall that in the training process, we minimize the log-likelihood loss of the fully connected network predictions which we denote by y_i where y_{gt} is the prediction score given to the correct ground truth class. Now, the total loss criteria with our regularization term can be written as

$$\mathcal{L} = -log\left(\frac{e^{y_{gt}}}{\sum_j e^{y_j}}\right) + \lambda\sum_{t=1}^{T}\|w'_t - w_t\|^2 \quad (1)$$

or as

$$\mathcal{L} = -log\left(\frac{e^{y_{gt}}}{\sum_j e^{y_j}}\right) + \lambda\sum_{t=1}^{T-1}\|w'_t - w_{t+1}\|^2 \quad (2)$$

where the first term in both (1) and (2) is the original classification loss. We call the second regularization term in (1) an *auto-encoder regularization* term and in (2) a *language model regularization* term. Intuitively, since each w'_t is obtained by a linear transformation of h_t, it enforces the hidden state h_t to maintain enough information on each w_t such it can be reconstructed back from h_t or such that the following word w_{t+1} can be predicted from h_t. This aids in obtaining a more general sentence representation and mitigates the risk of overfitting to the SNLI training set. The constant λ in (1) and (2) is a hyper-parameter that controls the amount of regularization and was set to 1 in our experiments.

We have also experimented with combining the two terms, giving equal weight to each of them in optimizing the model.

2.2 Bi-directional Regularization terms

Similarly to regularization terms described in 2.1, we devise variants of (1) and (2) which take into account the bi-directional architecture of the model. Here, we add two linear transformation layers: $\overrightarrow{L}_{\frac{l}{2}\times d} : \overrightarrow{H} \rightarrow W$ and $\overleftarrow{L}_{\frac{l}{2}\times d} : \overleftarrow{H} \rightarrow W$ on top of the forward LSTM and backward LSTM, respectively, and denote their output as $\{\overrightarrow{w}'_t\}$ and $\{\overleftarrow{w}'_t\}$, respectively, where $t = 1, \dots, T$.

Now, equations (1) and (2) are re-written as:

$$\mathcal{L} = -log\left(\frac{e^{y_{gt}}}{\sum_j e^{y_j}}\right) + \lambda_1\sum_{t=1}^{T}\|\overrightarrow{w}'_t - w_t\|^2 \quad (3)$$
$$+\lambda_2\sum_{t=1}^{T}\|\overleftarrow{w}'_t - w_t\|^2$$

and

$$\mathcal{L} = -log\left(\frac{e^{y_{gt}}}{\sum_j e^{y_j}}\right) + \lambda_1\sum_{t=1}^{T-1}\|\overrightarrow{w}'_t - w_{t+1}\|^2 \quad (4)$$
$$+\lambda_2\sum_{t=2}^{T}\|\overleftarrow{w}'_t - w_{t-1}\|^2$$

We call the second regularization term in (3) a *bi-directional auto-encoder regularization* and in (4) a *bi-directional language model regularization* term. Again, λ_1 and λ_2 are hyper-parameters controlling the amount of regularization and were set to 0.5 in our experiments.

3 Experiments

Following (Conneau et al., 2017) we have tested our approach on a wide array of classification tasks, including sentiment analysis (MR – Pang and Lee (2005), SST – Socher et al. (2013)), question-type (TREC – Li and Roth (2002)), product reviews (CR – Hu and Liu (2004)), subjectivity/objectivity (SUBJ – Pang and Lee (2005)) and opinion polarity (MPQA – Wiebe et al. (2005)). We also tested our approach on semantic textual similarity (STS 14 – Agirre et al. (2014)), paraphrase detection (MRPC – Dolan et al. (2004)), entailment and semantic relatedness tasks (SICK-R and SICK-E – Marelli et al. (2014)), though those tasks are more close in nature to the task of the SNLI dataset which the model was trained on. In our experiments we have set λ from eq. (1) and eq. (2) to be 1 and λ_1, λ_2 from eq. (3) and eq. (4) to be 0.5. All other hyper-parameters and implementation details were left unchanged to provide a fair comparison to the baseline method of (Conneau et al., 2017).

Our results are summarized in table 1. We compared out method against the baseline BiLSTM implementation of (Conneau et al., 2017) and included FastSent (Hill et al., 2016) and SkipThought vectors (Kiros et al., 2015) as a reference.

As evident from table 1 in almost all the tasks evaluated, adding the proposed regularization terms improves performance. This serve to show that in a supervised learning setting, additional information on the input sequence can be leveraged and injected to the model by adding simple unsupervised loss criteria.

4 Conclusions

In our work, we have sought to connect unsupervised and supervised learning in the context of sentence embeddings. Leveraging supervision given by some general task aided in obtaining state-of-the-art sentence representations (Conneau et al., 2017). However, every supervised learning tasks is prone to overfit. In this context, overfitting to the learning task will result in a model which generalizes less well to new tasks.

We alleviate this problem by incorporating unsupervised regularization criteria in the model's loss function which are motivated by autoencoders and language models. We note that the added regularization terms do come at the price of increasing the model size by ld parameters (where d and l are the dimensions of the word embedding and the LSTM hidden state, respectively) due to the added linear transformation (see 2.1). However, as evident from our results, this does not hinder the model performance, even though we did not increase the amount of training data. Moreover, since those term are unsupervised in nature, it is possible to pre-train the model on unlabeled data and then finetune it on the SNLI dataset.

In conclusion, our experiments show that adding the proposed regularization terms results in a more general model and superior sentence embeddings. This validates our assumption that while the a supervised signal is general enough for learning sentence embeddings, it can be further improved by incorporated a second unsupervised signal.

5 Acknowledgments

We would like to thank Shimi Salant and Ofir Press for their helpful comments.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*. pages 81–91.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* .

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 670–680.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, page 350.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of NAACL-HLT*. pages 1367–1377.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 168–177.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*. pages 3294–3302.

Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. 2015. Associating neural word embeddings with deep image representations using fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 4437–4446.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. pages 1188–1196.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pages 1–7.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *LREC*. pages 216–223.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*. pages 6297–6308.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*. volume 2, page 3.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 115–124.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1756–1765.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* .

Shimi Salant and Jonathan Berant. 2017. Contextualized word representations for reading comprehension. *arXiv preprint arXiv:1712.03609* .

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. pages 1631–1642.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation* 39(2-3):165–210.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* .