Contrasting distinct structured views to learn sentence embeddings

Antoine Simoulin[1,2] **Benoît Crabbé**[1]
[1]University of Paris, LLF [2]Quantmetry
asimoulin@quantmetry.com
benoit.crabbe@linguist.univ-paris-diderot.fr

Abstract

We propose a self-supervised method that builds sentence embeddings from the combination of diverse explicit syntactic structures of a sentence. We assume structure is crucial to building consistent representations as we expect sentence meaning to be a function of both syntax and semantic aspects. In this perspective, we hypothesize that some linguistic representations might be better adapted given the considered task or sentence. We, therefore, propose to learn individual representation functions for different syntactic frameworks jointly. Again, by hypothesis, all such functions should encode similar semantic information differently and consequently, be complementary for building better sentential semantic embeddings. To assess such hypothesis, we propose an original contrastive multi-view framework that induces an explicit interaction between models during the training phase. We make experiments combining various structures such as dependency, constituency, or sequential schemes. Our results outperform comparable methods on several tasks from standard sentence embedding benchmarks.

1 Introduction

We propose a self-supervised method that builds sentence embeddings from the combination of diverse explicit syntactic structures. The method aims at improving the ability of models to yield compositional sentence embeddings. We evaluate the embedding potential to solve downstream tasks.

Building generic sentence embeddings remains an open problem. Many training methods have been explored: generating past and previous sentences (Kiros et al., 2015; Hill et al., 2016), discriminating context sentences (Logeswaran and Lee, 2018), predicting specific relations between pairs of sentences (Conneau et al., 2017; Nie et al., 2019). While all these methods propose efficient training objectives, they all rely on a similar Recurrent Neural Network (RNN) as encoder architecture. Nonetheless, model architectures have been subject to extensive work as well (Tai et al., 2015; Zhao et al., 2015; Arora et al., 2017; Lin et al., 2017), and in supervised frameworks, many encoder structures outperform standard RNN networks.

We hypothesize structure is a crucial element to perform compositional knowledge. In particular, the heterogeneity of performances across models and tasks makes us assume that some structures may be better adapted for a given example or task. Therefore, combining diverse structures should be more robust for tasks requiring complex word composition to derive their meaning. Hence, we aim here to evaluate the potential benefit from interactions between pairs of encoders. In particular, we propose a training method for which distinct encoders are learned jointly. We conjecture this association might improve our embeddings' power of generalization and propose an experimental setup to corroborate our hypothesis.

We take inspiration from multi-view learning, which is successfully applied in a variety of domains. In such a framework, the model learns representations by aligning separate observations of the same object. Such observations are referred to as *views*. In our case, we consider a view for a given sentence as the association of the plain sentence with a syntactic structure.

As proposed in image processing (Tian et al., 2019; Bachman et al., 2019), we aim to align the different views using a contrastive learning framework. Indeed, contrastive learning is broadly used in NLP (Mikolov et al., 2013b,a; Logeswaran and Lee, 2018). We intend to enhance the sentence embedding framework proposed in Logeswaran and Lee (2018) with a multi-view paradigm.

Combining different structural views has already been proven to be successful in many NLP applica-

Proceedings of the 16th Conference of the European Chapter of the Associationfor Computational Linguistics: Student Research Workshop, pages 71–79
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

tions. Kong and Zhou (2011) provide a heuristic to combine dependency and constituency analysis for coreference resolution. Zhou et al. (2016); Ahmed et al. (2019) combine Tree LSTM and standard sequential LSTM with a cross-attention method and observe improvements on a semantic textual similarity task. Chen et al. (2017a) combine CNN and Tree LSTM using attention methods and outperform both models taken separately on a sentiment classification task. Finally, Chen et al. (2017b) combine sequential LSTM and Tree LSTM for natural language inference tasks.

The novelty here is to combine distinct structured models to build standalone sentence embeddings, which has not yet been explored. This paradigm benefits from several structural advantages. It pairs nicely with contrastive learning, as already mentioned. It might thus be trained in a self-supervised manner that does not require data annotation. Moreover, contrary to models presented in Section 2.2, our method is not specific to a certain kind of encoder architecture. It does not require, for example, the use of attention layers or tree-structured models. Our setup could therefore be extended with any encoding function. Finally, our training method induces an interaction between models during inference and, paramountly, during the training phase.

2 Method

Given a sentence s, the model aims at discriminating the sentences s^+ in the neighborhood of s from sentences s^- outside of this neighborhood. This is contrastive learning (Section 2.1). The representation of each sentence is acquired by using multiple views (Section 2.2).

2.1 Contrastive learning

Contrastive learning is successfully applied in a variety of domains including audio (van den Oord et al., 2018), image (Wu et al., 2018; Tian et al., 2019), video or natural language processing for word embedding (Mikolov et al., 2013b) or sentence embedding (Logeswaran and Lee, 2018). Some mathematical foundations are detailed in (Saunshi et al., 2019). The idea supposes to build a dataset such that each sample x is combined with another sample x^+, which is somehow *close*. For word or sentence embeddings, the close samples are the words or the sentences appearing in the given textual context. For image processing, close samples might be two different parts of the same image. Systems are trained to bring close samples together while dispersing negative examples.

In particular, a sentence embedding framework is proposed by Logeswaran and Lee (2018). The method takes inspiration from the distributional hypothesis successfully applied for word, but this time, to identify context sentences. The network is trained using a contrastive method. Given a sentence s, a corresponding context sentence s^+ and a set of K negative samples $s_1^- \cdots s_K^-$, the training objective is to maximize the probability of discriminate the correct sentence among negative samples: $p(s^+|S)$ with $S = \{s, s^+, s_1^- \cdots s_K^-\}$.

The algorithm architecture used to estimate p is close to *word2vec* (Mikolov et al., 2013b,a). As illustrated in Figure 1, two sentences encoders f and g are defined and the conditional probability is estimated as follow[1]:

$$p(s^+|S) = \frac{e^{c\left(f(s),g(s^+)\right)}}{e^{c(f(s),g(s^+))} + \sum_{i=1}^{N} e^{c\left(f(s),g(s_i^-)\right)}}$$

At inference time, the sentence representation is obtained as the concatenation of the two encoders f and g such as $s \rightarrow [f(s); g(s)]$, as illustrated in Figure 2. In Logeswaran and Lee (2018), f and g use the same RNN encoder. However, the authors observe that the encoders might learn redundant features. To limit this effect, they propose to use a distinct set of embeddings for each encoder.

We propose addressing this aspect by enhancing the method with a multi-view framework and using a distinct structured model for the encoders f and g. We hypothesize that some structures may be better adapted for a given example or task. For example, dependency parsing usually sets the verb as the root node. Whereas in constituency parsing, subject and verb are often the right and left child from the root node. Therefore, the combination of different structures should be more robust for tasks requiring complex word composition and be less sensitive to lexical variations. Consequently, we propose a training procedure that allows the model to benefit from the interaction of various syntactic structures. The choice for the encoder architecture is detailed in the following section.

[1] Logeswaran and Lee (2018) simply use an inner product for c such as $c(x, y) = x^T y$. In our case, as the encoders f and g are distincts, we choose a bilinear function defined as $c(x, y) = x^T W y$ (Tschannen et al., 2020).

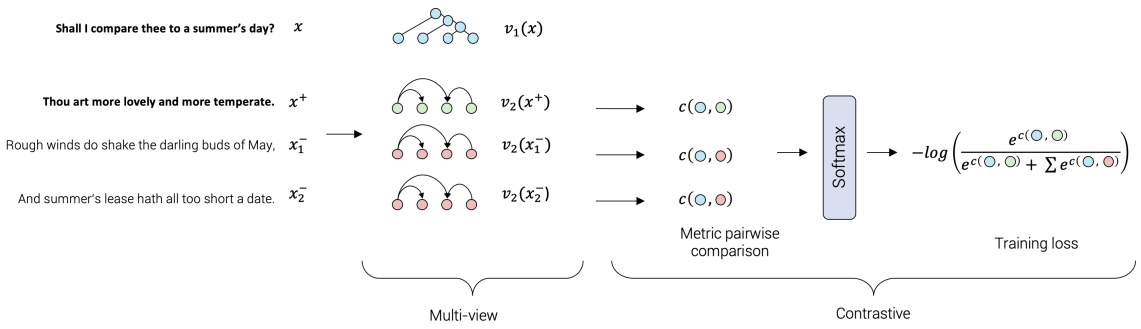

Figure 1: **Contrastive training method**. The objective is to reconstruct the storyline. Sentences are presented in their original order. Given an anchor sentence x, the model should identify the context sentence x^+ out of negative samples x_1^-, x_2^-. Sentences are encoded using separate views, which are composed within a pairwise distance matrix.

2.2 Language views

Multi-view aims as learning representations from data represented by multiple independent sets of features. As depicted in Section 1, we generalize the notion of view for a sentence as the application of a specific syntactic framework. For each view, we use an ad-hoc algorithm that maps the structured sentence into an embedding space.

We consider structures including sequence and trees detailed below. Although equivalences might be derived between the two representations schemes, we hypothesize that, in our context, the corresponding sequence of operations might allow capturing rather distinct linguistic properties. The various models may, therefore, be complementary and their combination allows for more fine-grained analysis.

Vanilla GRU (SEQ) assumes a sequential structure where each word depends on the previous words in the sentence. The framework is a bidirectional sequential GRU (Cho et al., 2014). The concatenation of the forward and backward last hidden state of the model is used as sequence embedding.

Dependency tree (DEP) In the dependency tree model, words are connected through dependency edges. A word might have an arbitrary number of dependents. The sentence can be represented as a tree where nodes corresponding to words and edges indicate whether or not the words are connected in the dependency tree. In our case, the dependency tree is obtained using the deep biaffine parser from Dozat and Manning (2017). The details of the parsing operations are detailed in Appendix A.1. For this view, we compute sentence embeddings with the Child-Sum Tree LSTM model described in Tai et al. (2015): Each node is assigned an embedding given its dependent with a recursive function. The recursive node function is derived from standard LSTM formulations but adapted for tree inputs. In particular, the hidden state is computed as the sum of all children hidden states. Here, we consider an Attentive Child-Sum Tree LSTM and we compute $\tilde{h}_j$ as the weighted sum of children vectors as in Zhou et al. (2016). The computation of $\tilde{h}_j$ in Equation 1 allows the model to filter semantically less relevant children.

$$\tilde{h}_j = \sum_{k \in C(j)} \alpha_{kj} h_k \tag{1}$$

With $C(j)$, the set of children of node j. All equations are detailed in Tai et al. (2015). The parameters α_{kj} are attention weights computed using a *soft attention layer*. Given a node j, we consider $h_1, h_2, \ldots, h_n$ the corresponding children hidden states. the soft attention layer produces a weight α_k for each child's hidden state. We did not use any external query to compute the attention but instead use a projection from the current node embedding. The attention equations are detailed below:

$$q_j = W^{(q)} x_j + b^{(q)}; \quad p_k = W^{(p)} h_k + b^{(p)} \tag{2}$$

$$a_{kj} = \frac{q_j \cdot p_k^\intercal}{\|q_j\|_2 \cdot \|p_k\|_2} \tag{3}$$

$$\alpha_{kj} = \text{softmax}_k(a_{1j} \cdots a_{nj}) \tag{4}$$

The embedding at the root of the tree is used as the sentence embedding as the Tree LSTM model computes representations bottom up.

Constituency tree (CONST) Constituent analysis describes the sentence as a nested multi-word

structure. In this framework, words are grouped recursively in constituents. In the resulting tree, only leaf nodes correspond to words, while internal nodes encode recursively word sequences. The structure is obtained using the constituency neural parser from Kitaev and Klein (2018). The framework is associated with the N-Ary Tree LSTM, which is defined in Tai et al. (2015). Similarly to the original article, we binarize the trees to ensure that every node has exactly two dependents. The binarization is performed using a left markovization and unary productions are collapsed in a single node. Again the representation is computed bottom-up and the embedding of the tree root node is used as sentence embedding. The equations detailed in Tai et al. (2015) make the distinction between right and left nodes. Therefore we do not propose to enhance the original architecture with a weighted sum as on the DEP view.

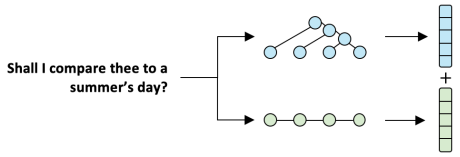

Figure 2: **Multi-view sentence embedding**. At inference, embeddings are the concatenation from both views.

3 Experiments

3.1 Training configuration

We train our models on the UMBC dataset [2,3] (Han et al., 2013). We limited our corpus to the first 40M sentences from the tokenized corpus. Indeed, Logeswaran and Lee (2018) already analyze the effect of the corpus size, and we focus here on the impact of our multi-view setting. We build batches from successive sentences. Given a sentence in a batch, other sentences not in the context are considered as negatives samples as presented in Section 2.1. Hyperparameters of the models such as the hidden size and the optimization procedure such as learning rate are detailed in Appendix A.2.

[2] https://ebiquity.umbc.edu/blogger/2013/05/01/umbc-webbase-corpus-of-3b-english-words/

[3] The bookcorpus introduced in Zhu et al. (2015) and traditionally used for sentence embedding is no longer distributed for copyright reasons. Therefore, we prefer a corpus freely available. The impact of the training dataset choice is analyzed in Appendix A.3.

3.2 Evaluation on downstream tasks

As usual for models aiming to build generic sentence embeddings (Kiros et al., 2015; Hill et al., 2016; Arora et al., 2017; Conneau et al., 2017; Logeswaran and Lee, 2018; Nie et al., 2019), we use tasks from the SentEval benchmark (Conneau and Kiela, 2018)[4]. SentEval is specifically designed to assess the quality of the embeddings themselves rather than the quality of a model specifically targeting a downstream task, as is the case for the GLUE and SuperGLUE benchmarks (Wang et al., 2019b,a). Indeed, the evaluation protocol prevents for fine-tuning the model during inference and the architecture to tackle the downstream tasks is kept minimal. Moreover, the embedding is kept identical for all tasks, thus assessing their properties of generalization.

Therefore, classification tasks from the SentEval benchmark are usually used for evaluation of sentence representations (Conneau and Kiela, 2018): the tasks include sentiment and subjectivity analysis (**MR, CR, SUBJ, MPQA**), question type classification (**TREC**), paraphrase identification (**MRPC**) and semantic relatedness (**SICK-R**). Contrasting the results of our model on this set of tasks will help to better understand its properties.

The MR, CR, SUBJ, MPQA tasks are binary classification tasks with no pre-defined train-test split. We therefore use a 10-fold cross validation. For the other tasks we use the proposed train/dev/test splits. We follow the linear evaluation protocol of Kiros et al. (2015), where a logistic regression or softmax classifier is trained on top of sentence representations. The dev set is used for choosing the regularization parameter and results are reported on the test set.

For the vocabulary, we follow the setup proposed in Kiros et al. (2015); Logeswaran and Lee (2018) and we train two models in each configuration. One initialized with pre-trained embedding vectors. The vectors are not updated during training and the vocabulary includes the top 2M cased words from the 300-dimensional GloVe vectors[5] (Pennington et al., 2014). The other is limited to 50K words initialized with a Xavier distribution and updated during training. For inference, the vocabulary is expanded to 2M words using a linear projection.

[4] Senteval is posterior to most of the references. However, these studies do evaluate on tasks later included in the benchmark.

[5] https://nlp.stanford.edu/projects/glove/

Model	Dim	Hrs	MR	CR	SUBJ	MPQA	TREC	MRPC Acc	MRPC F1	SICK-R r	SICK-R ρ	SICK-R MSE
Context sentences prediction												
FastSent	≤ 500	2	70.8	78.4	88.7	80.6	76.8	72.2	80.3	—	—	—
FastSent + AE	≤ 500	2	71.8	76.7	88.8	81.5	80.4	71.2	79.1	—	—	—
Skipthought	4800	336	76.5	80.1	93.6	87.1	92.2	73.0	82.0	85.8	79.2	26.9
Skipthought + LN	4800	672	79.4	83.1	93.7	89.3	—	—	—	85.8	78.8	27.0
Quickthoughts	4800	11	**80.4**	**85.2**	**93.9**	**89.4**	**92.8**	**76.9**	**84.0**	**86.8**	**80.1**	**25.6**
Sentence relations prediction												
InferSent	4096	—	**81.1**	**86.3**	92.4	90.2	88.2	**76.2**	**83.1**	**88.4**	—	—
DisSent Books 5	4096	—	80.2	85.4	93.2	90.2	91.2	76.1	—	84.5	—	—
DisSent Books 8	4096	—	79.8	85.0	**93.4**	**90.5**	**93.0**	76.1	—	85.4	—	—
Pre-trained transformers												
BERT-base [CLS]	768	96	78.7	84.9	94.2	88.2	**91.4**	71.1	—	75.7†	—	—
BERT-base [NLI]	768	96	**83.6**	**89.4**	**94.4**	**89.9**	89.6	**76.0**	—	**84.4**†	—	—
Our models (GloVe & Pretrained Embeddings)												
SEQ, CONST†	4800	41	79.8	82.9	94.6	88.5	90.4	76.4	83.7	86.1	78.9	26.3
DEP, SEQ†	4800	27	79.7	82.2	94.4	88.6	91.0	**77.9**	**84.4**	86.6	79.8	25.5
DEP, CONST†	4800	39	**80.7**	**83.6**	**94.9**	**89.2**	**92.6**	76.8	83.6	**87.0**	**80.3**	**24.8**

Table 1: **SentEval Task Results Using Fixed Sentence Encoder.** We divided the table into sections. The first range of models is directly comparable to our model as the training objective is to identify context sentences. The second section objective is to identify the correct relationship between a pair of sentences. The third section reports pre-trained transformers based-models. The last section reports the results from our models. FastSent is reported from Hill et al. (2016). Skipthoughts results from Kiros et al. (2015) Skipthoughts + LN which includes layer normalization method from Ba et al. (2016). We considered the Quickthoughts results (Logeswaran and Lee, 2018) with a pre-training on the bookcorpus dataset. DisSent and Infersent are reported from Nie et al. (2019) and Conneau et al. (2017) respectively. Pre-trained transformers results are reported from Reimers and Gurevych (2019). The **Hrs** column indicates indicative training time, the **Dim** column corresponds to the sentence embedding dimension. † indicates models that we had to re-train. Best results in each section are shown in **bold**, best results overall are underlined. Performance for **SICK-R** results are reported by convention as ρ and $r \times 100$.

3.3 Results analysis

We compare the properties of distinct views combination on downstream tasks. Results are compared with state of the art methods in Table 1. The first set of methods (*Context sentences prediction*) are trained to reconstruct books storyline. The second set of models (*Sentence relations prediction*) is pre-trained on a supervised task. Infersent (Conneau et al., 2017) is trained on the SNLI dataset, which proposes to predict the entailment relation between two sentences. DisSent (Nie et al., 2019) proposes a generalization of the method and builds a corpus of sentence pairs with more possible relations between them. Finally, we include models relying on transformer architectures (Pre-trained transformers) for comparison. In particular, BERT-base model and a BERT-model fine-tuned on the SNLI dataset (Reimers and Gurevych, 2019). In Table 1, we observe that our models expressing a combination of views such as (DEP, SEQ) or (DEP, CONST) give better results than the use of the same view (SEQ, SEQ) used in Quick-Thought model. It seems that the entanglement of views benefits the sentence embedding properties. In particular, we obtain state-of-the-art results for almost every metric from **MRPC** and **SICK-R** tasks, which focus on paraphrase identification. For the **MRPC** task, we gain a full point in accuracy and outperform BERT models. We hypothesize structure is important for achieving this task, especially as the dataset is composed of rather long sentences. The **SICK-R** dataset is structurally designed to discriminate models that rely on compositional operations.

This also explains the score improvement on this task. Tasks such as **MR**, **CR** or **MPQA** consist in sentiment or subjectivity analysis. We hypothesize that our models are less relevant in this case: such tasks are less sensitive to structure and depend more on individual word or lexical variation.

3.4 Impact of the multi-view

We aim to measure the impact of multi-view specifically. Table 2 compares all possible view pairs out

of DEP, CONST and SEQ views. For each multi-view model, we report the average score from SentEval tasks[6]. The first section of the Table corresponds to *single-view* models, for which both views from the pair are identical. The second section reports multi-view models.

Multi-view models outperform those using a single view. Given our experiment, it is advantageous to use multiple views instead of one. It also confirms our hypothesis that combining multiple structured models or views yield richer sentence embeddings.

Model	Avg. SentEval Score
Single-view models	
CONST, CONST	84.4
DEP, DEP	84.6
SEQ, SEQ	84.9
Multi-view models	
SEQ, CONST	85.1
SEQ, DEP	85.3
DEP, CONST	**86.0**

Table 2: **Impact of the multi-view.** The first section corresponds to single-view setups for which f and g are the same views. The second section reports multi-view models. For each model, we report the average score on the SentEval benchmark.

4 Conclusion and future work

Inspired from linguistic insights and supervised learning, we hypothesize that structure is a central element to build sentence embeddings. The novelty here is detailed in Section 2 and consists in jointly learning structured models in a contrastive framework. In Section 3 we evaluate the standalone sentence embeddings and use them as a feature for the dedicated SentEval benchmark. We obtain state-of-the-art results on tasks which are expected, by hypothesis, to be more sensitive to sentence structure. We show in Section 3.4 that multi-view embeddings yield better downstream task results. Our setup confirms our hypothesis that combining diverse structures should be more robust for tasks requiring to perform complex compositional knowledge.

[6]We scale all metrics as percentages. In particular, we use 100 - MSE for the **SICK-R** task. The final score corresponds to the average of all tasks. We average the scores for tasks with multiple metrics (**MRPC** and **SICK-R**).

References

Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E. Mercer. 2019. Improving tree-lstm with tree attention. In *13th IEEE International Conference on Semantic Computing, ICSC 2019, Newport Beach, CA, USA, January 30 - February 1, 2019*, pages 247–254.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Philip Bachman, R. Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 15509–15519.

Liu Chen, Guangping Zeng, Qingchuan Zhang, and Xingyu Chen. 2017a. Tree-lstm guided attention pooling of DCNN for semantic sentence modeling. In *5G for Future Wireless Networks - First International Conference, 5GWN 2017, Beijing, China, April 21-23, 2017, Proceedings*, pages 52–59.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1657–1668.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*.

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, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 670–680.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Steffen Eger, Andreas Rücklé, and Iryna Gurevych. 2019. Pitfalls in the evaluation of sentence embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2019, Florence, Italy, August 2, 2019*, pages 55–60. Association for Computational Linguistics.

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. Umbc_ebiquity-core: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA*, pages 44–52.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1367–1377.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3294–3302.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2676–2686.

Fang Kong and Guodong Zhou. 2011. Combining dependency and constituent-based syntactic information for anaphoricity determination in coreference resolution. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation, PACLIC 25, Singapore, December 16-18, 2011*, pages 410–419.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Allen Nie, Erin Bennett, and Noah Goodman. 2019. Dissent: Learning sentence representations from explicit discourse relations. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4497–4510.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. 2019. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5628–5637.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1556–1566.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive multiview coding. *CoRR*, abs/1906.05849.

Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. 2020. On mutual information maximization for representation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

John Wieting and Douwe Kiela. 2019. No training required: Exploring random encoders for sentence classification. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance-level discrimination. *CoRR*, abs/1805.01978.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 4069–4076.

Yao Zhou, Cong Liu, and Yan Pan. 2016. Modelling sentence pairs with tree-structured attentive encoder. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2912–2922.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27.

A Appendices

A.1 Parsing procedure

We use an open-source implementation[7] of the dependency parser (Dozat and Manning, 2017) and replace the pos-tags features with features obtained with BERT. Therefore we do not need pos-tags annotations to parse our corpus. Regarding the inference speed, The constituency parser is the bottleneck in this case and parse around 500 sentences/second. In our case, the parsing of the entire corpus (40M sentences) take about a day to complete. Regarding the model, we implemented tree models using an efficient batching method which allows us to keep training in a reasonable range (maximum 41 hours.)

A.2 Hyper parameters

Model hyper parameters are fixed given literature on comparable work (Tai et al., 2015; Logeswaran and Lee, 2018). All models are trained using a batch size of 400 and the Adam optimizer with a $5e^{-4}$ learning rate. Regarding the infrastructure, we use a Nvidia GTX 1080 Ti GPU. All model weights are initialized with a Xavier distribution and biases set to 0. We do not apply any dropout.

A.3 Impact of the training dataset

We train our model on the UMBC dataset. We have chosen to make use of a distinct corpus as the BookCorpus dataset is no longer distributed for copyright reasons. We have run QuickThought scripts (Logeswaran and Lee, 2018) using our dataset based on the UMBC corpus to compare both setups. Results are detailed in the first Section from Table 3 and are rather close in both configurations. Indeed, except for the **SUBJ** and **MR** task, the use of our dataset penalizes the results. Our corpus is indeed restricted to 40M sentences, in comparison with 74M for the Bookcorpus. Regarding the dataset size and the SentEval results, we have considered the comparison holds.

A.4 Biases toward embedding size

SentEval evaluation framework is suspected to suffers from biases toward the embedding size (Eger et al., 2019). Moreover, some works on sentence embedding evaluation methods points surprising good results may be achieved using randomly initialized encoders (Wieting and Kiela, 2019). We

[7] https://github.com/yzhangcs/biaffine-parser

Model	MR	CR	SUBJ	MPQA	TREC	MRPC Acc	MRPC F1	SICK-R r	SICK-R ρ	SICK-R MSE
Impact of the pretraining corpus on QuickThought										
Quickthoughts (results from paper)	80.4	**85.2**	93.9	**89.4**	**92.8**	**76.9**	84.0	**86.8**	**80.1**	**25.6**
Quickthoughts (UMCB 40M)†	**80.9**	84.4	**95.1**	88.9	92.2	75.8	—	86.0	—	—
Impact of the embedding size										
BERT-base [CLS] †	**77.3**	81.3	92.7	85.0	80.2	69.9	—	61.0	—	—
BERT-base [CLS] /w random projection †	77.1	**82.6**	**93.1**	**85.9**	**80.8**	**71.3**	—	**71.0**	—	—
Impact of pre-training										
DEP, CONST†	**80.7**	**83.6**	**94.9**	**89.2**	**92.6**	**76.8**	**83.6**	**87.0**	**80.3**	**24.8**
Rand LSTM	77.2	78.7	91.9	87.9	86.5	74.1	—	86.0	—	—

Table 3: **Ablation study on SentEval task results.** The first section compares the impact of the training dataset for QuickToughts. The next section focuses on the impact of the embedding size. To this end, hidden representations are projected into a larger embedding space using a random, fully connected layer. The final Section compares models randomly initialized with those pre-trained on our self-supervised task. † indicates models that we had to re-train.

provide extra analysis to discuss these potential pitfalls.

Regarding the dependency on the embedding size, we run experiments to analyze if such bias could explain BERT low performances on SentEval since the output hidden size is only of 768. Following the protocol from Wieting and Kiela (2019), we project the embedding from the CLS token using a random matrix initialized with a glorot distribution. This setup expands BERT embedding into 4096 dimensions. We reported the results in Table 3. We observe expanding the embedding size seems to slightly improve the results. However, the results are still below Quickthought vectors by a large margin.

Regarding the effect of randomly initialized encoders (Wieting and Kiela, 2019), we reported the results in Table 3. Although randomly initialized encoders achieve surprisingly good results, they are still below our results obtained with pre-training.