Sentence Similarity Based on Contexts

Xiaofei Sun[♦], Yuxian Meng[♣], Xiang Ao[▲], Fei Wu[♦], Tianwei Zhang[♥], Jiwei Li[♦♣], and Chun Fan[♠]

[♦]Zhejiang University, China, [♣]Shannon.AI, China, [▲]Chinese Academy of Sciences, China,
[♥]Nanyang Technological University, Singapore, [♠]Computer Center, Peking University, China,
[♠]National Biomedical Imaging Center, Peking University, China, [♠]Peng Cheng Laboratory, China
{xiaofei_sun,yuxian_meng,jiwei_li}@shannonai.com,aoxiang@ict.ac.cn
wufei@zju.edu.cn,tianwei.zhang@ntu.edu.sg,fanchun@pku.edu.cn

Abstract

Existing methods to measure sentence similarity are faced with two challenges: (1) labeled datasets are usually limited in size, making them insufficient to train supervised neural models; and (2) there is a training-test gap for unsupervised language modeling (LM) based models to compute semantic scores between sentences, since sentence-level semantics are not explicitly modeled at training. This results in inferior performances in this task. In this work, we propose a new framework to address these two issues. The proposed framework is based on the core idea that the meaning of a sentence should be defined by its contexts, and that sentence similarity can be measured by comparing the probabilities of generating two sentences given the same context. The proposed framework is able to generate high-quality, large-scale dataset with semantic similarity scores between two sentences in an unsupervised manner, with which the train-test gap can be largely bridged. Extensive experiments show that the proposed framework achieves significant performance boosts over existing baselines under both the supervised and unsupervised settings across different datasets.

1 Introduction

Measuring sentence similarity is a long-standing task in NLP (Luhn, 1957; Robertson et al., 1995; Blei et al., 2003; Peng et al., 2020). The task aims at quantitatively measuring the semantic relatedness between two sentences, and has wide applications in text search (Farouk et al., 2018), natural language understanding (MacCartney and Manning, 2009), and machine translation (Yang et al., 2019a).

One of the greatest challenges that existing methods face for sentence similarity is the lack of large-scale labeled datasets, which contain sentence pairs with labeled semantic similarity scores. The acquisition of such a dataset is both labor-intensive and expensive. For example, the STS benchmark (Cer et al., 2017) and SICK-Relatedness dataset (Marelli et al., 2014) respectively contain 8.6K and 9.8K labeled sentence pairs, the sizes of which are usually insufficient for training deep neural networks.

Unsupervised learning methods are proposed to address this issue, where word embeddings (Le and Mikolov, 2014) or BERT embeddings (Devlin et al., 2018) are used to to map sentences to fix-length vectors in an unsupervised manner. Then sentence similarity is computed based on the cosine or dot product of these sentence representations. Our work follows this thread where sentence similarity is computed based on fix-length sentence representations, as opposed to comparing sentences directly. The biggest issue with current unsupervised approaches is that there exists a big gap between model training and testing (i.e., computing semantic similarity between two sentences). For example, the BERT-style models are trained at the token level by predicting words given contexts, and there is neither explicit modeling sentence semantics nor producing sentence embeddings at the training stage. But at test time, sentence semantics needs to be explicitly modeled to obtain semantic similarity. The inconsistency results in a distinct discrepancy between the objectives at the two stages and inferior performance on textual semantic similarity tasks. For example, BERT embeddings yield inferior performance on semantic similarity benchmarks (Reimers and Gurevych, 2019), and even underperform the naive method such as averaging GloVe (Pennington et al., 2014) embeddings.

Transactions of the Association for Computational Linguistics, vol. 10, pp. 573–588, 2022. https://doi.org/10.1162/tacl_a_00477
Action Editor: Chris Quirk. Submission batch: 6/2021; Revision batch: 11/2021; Published 5/2022.

Li et al. (2020) investigated this problem and found that BERT always induces a non-smooth anisotropic semantic space of sentences, and this property significantly harms the performance of semantic similarity.

Just as word meanings are defined by neighboring words (Harris, 1954), the meaning of a sentence is determined by its contexts. Given the same context, there is a high probability of generating two similar sentences. If there is a low probability of generating two sentences given the same context, there is a gap between these two sentences in the semantic space. Based on this idea, we propose a framework that measures semantic similarity through the probability similarity of generating two sentences given the same context in a fully unsupervised manner. As for implementation, the framework consists of the following steps: (1) we train a contextual model by predicting the probability of a sentence fitting into the left and right contexts; (2) we obtain sentence pair similarity by comparing scores assigned by the contextual model across a large number of contexts. To facilitate inference, we train a surrogate model, to act as the role of step 2, based on the outputs from step 1. The surrogate model can be directly used for sentence similarity prediction in an unsupervised setup, or used as initialization to be further finetuned on downstream datasets in the supervised setup. Note that the outcome from step 1 or the surrogate model is a fixed-length vector regarding the input sentence. Each element in the vector indicates how fit the input sentence is to the context corresponding to that element, and the vector itself can be viewed as the overall semantics of the input sentence in the contextual space. Then we use cosine distance between two sentence vectors to compute the semantic similarity.

The proposed framework offers the potential to fully address the two challenges above: (1) the context regularization provides a reliable means to generate a large-scale high-quality dataset with semantic similarity scores based on unlabeled corpus; and (2) the train-test gap can be naturally bridged by training the model on the large-scale similarity dataset, leading to significant performance gains compared to utilize pretrained models directly.

We conduct experiments on different datasets under both supervised and unsupervised setups, and experimental results show that the proposed framework significantly outperforms existing sentence similarity models.

2 Related Work

Statistics-based methods for measuring sentence similarity include bag-of-words (BoW) (Li et al., 2006), term frequency inverse document frequency (TF-IDF) (Luhn, 1957; Jones, 2004), BM25 (Robertson et al., 1995), latent semantic indexing (LSI) (Deerwester et al., 1990), and latent Dirichlet allocation (LDA) (Blei et al., 2003). Deep learning based methods for sentence similarity rely on distributed representations (Mikolov et al., 2013; Le and Mikolov, 2014) and can be generally divided into the following three categories.

Matrix Based Methods

The first line of work for measuring sentence similarity is to construct a similarity matrix between two sentences, each element of which represents the similarity between the two corresponding units in two sentences. Then the matrix is aggregated in different ways to induce the final similarity score. Pang et al. (2016) applied a two-layer convolutional neural network (CNN) followed by a feed-forward layer to the similarity matrix to derive the similarity score. He and Lin (2016) used a deeper CNN to make the best use of the similarity matrix. Yin and Schütze (2015) built a hierarchical architecture to model text compositions at different granularities, so several similarity matrices can be computed and combined for interactions. Other works proposed using the attention mechanism as a way of computing the similarity matrix (Rocktäschel et al., 2015; Wang et al., 2016; Parikh et al., 2016; Seo et al., 2016; Shen et al., 2017; Lin et al., 2017; Gong et al., 2017; Tan et al., 2018; Kim et al., 2019; Yang et al., 2019b).

Word Distance Based Methods

The second line of work to measure sentence similarity is to calculate the cost of transforming from one sentence to another; the smaller the cost is, the more similar two sentences are. This idea is implemented by the Word Mover's Distance (WMD) (Kusner et al., 2015), which measures the dissimilarity between two documents as the minimum amount of distance that the embedded words of one document need to transform to words of another document. Following works improve WMD

by incorporating supervision from downstream tasks (Huang et al., 2016), introducing hierarchical optimal transport over topics (Yurochkin et al., 2019), addressing the complexity limitation of requiring to consider each pair (Wu and Li, 2017; Wu et al., 2018; Backurs et al., 2020), and combining graph structures with WMD to perform cross-domain alignment (Chen et al., 2020). More recently, Yokoi et al. (2020) proposed to disentangle word vectors in WRD have shown significant performance boosts over vanilla WMD.

Sentence Embedding Based Methods

Sentence embeddings are high-dimensional representations for sentences. They are expected to contain rich sentence semantics so that the similarity between two sentences can be computed by considering their sentence embeddings via certain metrics such as cosine similarity. Le and Mikolov (2014) introduced paragraph vector, which is learned in an unsupervised manner by predicting the words within the paragraph using the paragraph vector. In a followup, a line of sentence embedding methods such as FastText, Skip-Thought vectors (Kiros et al., 2015), Smooth Inverse Frequency (SIF) (Arora et al., 2017), Sequential Denoising Autoencoder (SDAEs) (Hill et al., 2016), InferSent (Conneau et al., 2017), Quick-Thought vectors (Logeswaran and Lee, 2018), and Universal Sentence Encoder (Cer et al., 2018) have been proposed to improve the sentence embedding quality with more efficiency.

The great success achieved by large-scale pretraining models (Devlin et al., 2018; Liu et al., 2019) has recently stimulated a strand of work on producing sentence embeddings based on the pretraining-finetuning paradigm using large-scale unlabeled corpora. The cosine outcome between the representations of two sentences produced by large-scale pretrained models is treated as the semantic similarity (Reimers and Gurevych, 2019; Wang and Kuo, 2020; Li et al., 2020). Su et al. (2021) and Huang et al. (2021) proposed regularizing the sentence representations by whitening them, that is, enforcing the covariance to be an identity matrix to address the non-smooth anisotropic distribution issue (Li et al., 2020).

The BERT-based scores (Zhang et al., 2020; Sellam et al., 2020), though serving as automatic metrics, also capture rich semantic information regarding the sentence and have the potentials for measuring semantic similarity. Cer et al. (2018) proposed a method of encoding sentences into their corresponding embeddings that specifically target transfer learning to other NLP tasks. Karpukhin et al. (2020) adopted two unique BERT encoder models and the model weights are optimized to maximize the dot product. The most recent line of work focuses on leveraging the contrastive learning framework to tackle semantic textual similarity (Wu et al., 2020; Carlsson et al., 2021; Kim et al., 2021; Yan et al., 2021; Gao et al., 2021), where two similar sentences are pulled close and two random sentences are pulled away in the sentence representation space. This learning strategy helps better separate sentences with different semantics.

This work is motivated by learning word representations given its contexts (Mikolov et al., 2013; Le and Mikolov, 2014) with the assumption that the meaning of a word is determined by its context. Our work is based on large-scale pretrained model and aims at learning informative sentence representations for measuring sentence similarity.

3 Model

3.1 Overview

The key point of the proposed paradigm is to compute semantic similarity between two sentences by measuring the probabilities of generating the two sentences across a number of context.

We can achieve this goal based on the following steps: (1) we first need to train a contextual model to predict the probability of a sentence fitting into the left and right contexts. This goal can be achieved by either a discriminative model, namely, predicting the probability that the concatenation of a sentence with context forms a coherent text, or a generative model, namely, predicting the probability of generating a sentence given contexts; (2) next, given a pair of sentences, we can measure their similarity by comparing their scores assigned by contextual models given different contexts; (3) for step 2, for any pair of sentences at test time, we need to sample different contexts to compute scores assigned by contextual models, which is time-consuming. We thus propose to train a surrogate model that takes a pair of sentences as inputs and predicts the similarity assigned by the contextual model. This enables faster inference, though at a small sacrifice of accuracy; (4) the surrogate

model can be directly used for obtaining sentence similarity scores in a unsupervised manner, or used as model initialization, which will be further fine-tuned on downstream datasets in a supervised setting. We will discuss the detail of each module in order below.

3.2 Training Contextual Models

We need a contextual model to predict the probability of a sentence fitting into left and right contexts. We combine a generative model and a discriminative model to achieve this goal, allowing us to take the advantage of both to model text coherence (Li et al., 2017).

Notations Let $\boldsymbol{c}_i$ denote the i-th sentence, which consists of a sequence of words $\boldsymbol{c}_i = \{c_{i,1}, \ldots, c_{i,n_i}\}$, where n_i denotes the number of words in $\boldsymbol{c}_i$. Let $\boldsymbol{c}_{i:j}$ denote the i-th to j-th sentences. $\boldsymbol{c}_{<i}$ and $\boldsymbol{c}_{>i}$ respectively denote the preceding and subsequent context of $\boldsymbol{c}_i$.

3.2.1 Discriminative Models

The discriminative model takes a sequence of consecutive sentences $[\boldsymbol{c}_{<i}, \boldsymbol{c}_i, \boldsymbol{c}_{>i}]$ as the input, and maps the input to a probability indicating whether the input is natural and coherent. We treat sentence sequences taken from the original articles written by humans as positive examples and sequences with replacements of the center sentence $\boldsymbol{c}_i$ as negative ones. Half of replacements of $\boldsymbol{c}_i$ come from the original document, and half of replacements come from random sentences from the corpus. The concatenation of LSTM representations at the last step (right-to-left and left-to-right) is used to represent the sentence. Sentence representations for consecutive sentences are concatenated and output to the sigmoid function to obtain the final probability:

$$p(y=1|\boldsymbol{c}_i, \boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}) = \text{sigmoid}(\boldsymbol{h}^\top [\boldsymbol{h}_{<i}, \boldsymbol{h}_i, \boldsymbol{h}_{>i}]) \quad (1)$$

where $\boldsymbol{h}$ denotes learnable parameters. We deliberately make the discriminative model simple for two reasons: The discriminative approach for coherence prediction is a relatively easy task and more importantly, it will be further used in the next selection stage for screening, where faster speed is preferred.

3.2.2 Generative Models

Given contexts $\boldsymbol{c}_{<i}$ and $\boldsymbol{c}_{>i}$, the generative model predicts the probability of generating each token in sentence $\boldsymbol{c}_i$ sequentially using SEQ2SEQ structures (Sutskever et al., 2014) as the backbone:

$$p(\boldsymbol{c}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}) = \prod_j p(\boldsymbol{c}_{i,j}|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}, \boldsymbol{c}_{i,<j}) \quad (2)$$

Semantic similarity between two sentences can be measured by not only the forward probability of generating the two sentences given the same context $p(\boldsymbol{c}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$, but also the backward probability of generating contexts given sentences. The context-given-sentence probability can be modeled by predicting preceding contexts given subsequent contexts $p(\boldsymbol{c}_{<i}|\boldsymbol{c}_i, \boldsymbol{c}_{>i})$ and to predict subsequent contexts given preceding contexts $p(\boldsymbol{c}_{>i}|\boldsymbol{c}_{<i}, \boldsymbol{c}_i)$.

3.3 Scoring Sentence Pairs

Given context $[\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}]$, the score for $\boldsymbol{s}_i$ fitting into the context is the linear combination of scores from discriminative and generative models:

$$\begin{aligned} S(\boldsymbol{s}_i, \boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}) &= \lambda_1 \log p(y=1|\boldsymbol{s}_i, \boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}) \\ &+ \lambda_2 \frac{1}{|\boldsymbol{s}_i|} \log p(\boldsymbol{s}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}) \\ &+ \lambda_3 \frac{1}{|\boldsymbol{c}_{<i}|} \log p(\boldsymbol{c}_{<i}|\boldsymbol{s}_i, \boldsymbol{c}_{>i}) \\ &+ \lambda_4 \frac{1}{|\boldsymbol{c}_{>i}|} \log p(\boldsymbol{c}_{>i}|\boldsymbol{c}_{<i}, \boldsymbol{s}_i) \end{aligned} \quad (3)$$

where λ_1, λ_2, λ_3, λ_4 control the tradeoff between different modules. For simplification, we use $\boldsymbol{c}$ to denote context $\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}$. $S(\boldsymbol{s}_i, \boldsymbol{c})$ is thus equivalent to $S(\boldsymbol{s}_i, \boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$.

Let $\boldsymbol{C}$ denote a set of contexts, where $N_{\boldsymbol{C}}$ is the size of $\boldsymbol{C}$. For a sentence $\boldsymbol{s}$, its semantic representation $\boldsymbol{v}_{\boldsymbol{s}}$ is an $N_{\boldsymbol{C}}$ dimensional vector, with each individual value being $S(\boldsymbol{s}, \boldsymbol{c})$ with $\boldsymbol{c} \in \boldsymbol{C}$. The semantic similarity between two sentences $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ can be computed based on $\boldsymbol{v}_{\boldsymbol{s}_1}$ and $\boldsymbol{v}_{\boldsymbol{s}_2}$ using different metrics such as cosine similarity.

Constructing $\boldsymbol{C}$ We need to pay special attentions to the construction of $\boldsymbol{C}$. The optimal situation is to use all contexts, where $\boldsymbol{C}$ is the entire corpus. Unfortunately, this is computationally prohibitive as we need to iterate over the entire corpus for each sentence $\boldsymbol{s}$.

We propose the following workaround for tractable computation. For a sentence $\boldsymbol{s}$, rather than using the full corpus as $\boldsymbol{C}$, we construct its sentence specific context set $\boldsymbol{C}_{\boldsymbol{s}}$ in a way that s can fit into all constituent context in $\boldsymbol{C}_{\boldsymbol{s}}$. The

intuition is as follows. With respect to sentence $\boldsymbol{s}_1$, contexts can be divided into two categories: contexts that $\boldsymbol{s}_1$ fits into, based on which we will measure whether or not $\boldsymbol{s}_2$ also fits in, and contexts that $\boldsymbol{s}_1$ does not fit into, and we will measure whether or not $\boldsymbol{s}_2$ also does not fit in. We are mostly concerned about the former, and can neglect the latter. The reason is as follows: The latter can also further be divided into two categories: contexts that fit neither $\boldsymbol{s}_1$ or $\boldsymbol{s}_2$, and contexts that do not fit $\boldsymbol{s}_1$ but fit $\boldsymbol{s}_2$. For contexts that fit neither $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, we can neglect them since two sentences not fitting into the same context does not signify their semantic relatedness; for contexts that does not fit $\boldsymbol{s}_1$ but fit $\boldsymbol{s}_2$, we can leave them to when we compute $C_{\boldsymbol{s}_2}$.

Practically, for a given sentence $\boldsymbol{s}$, we first use TF-IDF weighted BoW bi-gram vectors to perform primary screening on the whole corpus to retrieve related text chunks (20K for each sentence). Next, we rank all contexts using the discriminative model based on Eq. (1). For discriminative models, we cache sentence representations in advance, and compute model scores in the last neural layer, which is significantly faster than the generative model. This two-step selection strategy is akin to the pipelined selection system (Chen et al., 2017; Karpukhin et al., 2020) in open-domain QA that contains document retrieval using IR systems and fine-grained question answering using neural QA models.

$\boldsymbol{C_s}$ is built by selecting top ranked contexts by Eq. (3). We use the incremental construction strategy, adding one context at a time. To promote diversity of $\boldsymbol{C_s}$, each text chunk is allowed to contribute at most one context, and the Jaccard similarity between the $i-1$-th sentence in the context to select and those already selected should be lower than 0.5.[1]

To compute semantic similarity between $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, we concatenate $\boldsymbol{C}_{\boldsymbol{s}_1}$ and $\boldsymbol{C}_{\boldsymbol{s}_2}$ and use the concatenation as the context set $\boldsymbol{C}$. The semantic similarity score between $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ is given as follows:

$$\begin{aligned} \boldsymbol{v}_{\boldsymbol{s}_1} &= [S(\boldsymbol{s}_1, \boldsymbol{c}) \ \text{ for } \boldsymbol{c} \in \boldsymbol{C}_{\boldsymbol{s}_1} + \boldsymbol{C}_{\boldsymbol{s}_2}] \\ \boldsymbol{v}_{\boldsymbol{s}_2} &= [S(\boldsymbol{s}_2, \boldsymbol{c}) \ \text{ for } \boldsymbol{c} \in \boldsymbol{C}_{\boldsymbol{s}_1} + \boldsymbol{C}_{\boldsymbol{s}_2}] \\ \text{sim}(\boldsymbol{s}_1, \boldsymbol{s}_2) &= \text{cosine}(\boldsymbol{v}_{\boldsymbol{s}_1}, \boldsymbol{v}_{\boldsymbol{s}_2}) \end{aligned} \tag{4}$$

[1]This strategy can also remove text duplicates.

3.4 Training Surrogate Models

The method described in Section 3.3 provides a direct way to compute scores for semantic relatedness. But it comes with a severe shortcoming of slow speed at inference time: Given an arbitrary pair of sentences, the model still needs to go through the entire corpus, harvest the context set $\boldsymbol{C_s}$, and iterate all instances in $\boldsymbol{C_s}$ for context score calculation based on Eq. (3), each of which is time consuming. To address this issue, we propose training a surrogate model to accelerate inference.

Specifically, we first harvest similarity scores for sentence pairs using methods in Section 3.3. We collect scores for 100M pairs in total, which are further split into train/dev/test by 98/1/1. Next, by treating harvested similarity scores as gold labels, we train a neural model that takes a pair of sentence as an input, and predicts its similarity score. The cosine similarity between the two sentence representations is the predicted semantic similarity, and we minimize the L_2 distance between predicted and golden similarities. The Siamese structure makes it possible for fixed-sized vectors for input sentences to be derived and stored, allowing for fast semantic similarity search, which we will discuss in detail in the ablation study section.

It is worth noting both the advantages and disadvantages of the surrogate model. For advantages, firstly, it can significantly speed up inference as it avoids the time-consuming process of iterating over the entire corpus to construct C. Secondly, the surrogate shares the same structure with existing widely-used models such as BERT and RoBERTa, and can thus later be easily fine-tuned on the human-labeled datasets in supervised learning; on the other hand, the origin model in Section 3.3 cannot be readily combined with other human-labeled datasets. For disadvantages, the surrogate model inevitably comes with a cost of accuracy, as its upper bound is the origin model in Section 3.3.

4 Experiments

4.1 Experiment Settings

We evaluate the *Surrogate* model on Semantic Textual Similarity (STS), Argument Facet Similarity (AFS) corpus (Misra et al., 2016), and Wikipedia Sections Distinction (Ein Dor et al., 2018) tasks. We perform both unsupervised and

supervised evaluations on these tasks. For unsupervised evaluations, models are directly used for obtaining sentence representations. For supervised evaluations, we use the training set to fine-tune all models and use the L_2 regression as the objective function. Additionally, we also conduct partially supervised evaluation on STS benchmarks.

Implementation Details For discriminative model in 3.2.1, we use a single-layer bi-directional LSTM as the backbone with the size of hidden states set to 300.

For the generative model in 3.2.2, we implement the above three models, namely, $p(\boldsymbol{c}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$, $p(\boldsymbol{c}_{<i}|\boldsymbol{c}_i, \boldsymbol{c}_{>i})$, and $p(\boldsymbol{c}_{>i}|\boldsymbol{c}_{<i}, \boldsymbol{c}_i)$ based on the SEQ2SEQ structure, and use Transformer-large as the backbone (Vaswani et al., 2017). Sentence position embeddings and token position embeddings are added to word embeddings. The model is trained on a corpus extracted from CommonCrawl that contains 100B tokens.

For the surrogate model in 3.4, we use RoBERTa (Liu et al., 2019) as the backbone, and adopt the Siamese structure (Reimers and Gurevych, 2019), where two sentences are first mapped to vector representations using RoBERTa. We use the average pooling on the last RoBERTa layer to obtain the sentence representation. During training, we use Adam (Kingma and Ba, 2014) with learning rate of 1e-4, β_1 = 0.9, β_2 = 0.999. The trained surrogate model obtains an average L_2 distance of 7.4×10^{-4} on dev set when trained from scratch, and 6.1×10^{-4} when initialized using the RoBERTa-large model (Liu et al., 2019). We set $\boldsymbol{C_s}$ to 500.

Baselines We use the following models as baselines:

- **Avg. Glove embeddings** is the average of word embeddings produced via the co-occurrence statistics in the corpus (Pennington et al., 2014).
- **Avg. Skip-Thought embeddings** is the average of word embeddings produced by Skip-Thought vectors (Kiros et al., 2015).
- **InferSent** uses a Siamese BiLSTM network with max-pooling over the output on NLI datasets (Conneau et al., 2017).
- **Avg. BERT embeddings** is the average of word embeddings produced by BERT (Devlin et al., 2018).
- **BERT [CLS]** computes scores based on the vector representation of the special token [CLS] in BERT.
- **BERTScore** computes the similarity of two sentences as a sum of cosine similarities between their tokens' embeddings (Zhang et al., 2020).
- **BLEURT** is baseed on BERT and captures non-trivial semantic similarities by fine-tuning the model on the WMT Metrics dataset, on a set of ratings provided by the user, or a combination of both (Sellam et al., 2020).
- **DPR** works by using two unique BERT encoder models and the model weights are optimized to maximize the dot product (Karpukhin et al., 2020).
- **Universal Sent Encoder** is a method of encoding sentences into their corresponding embeddings that specifically target transfer learning to other NLP tasks (Cer et al., 2018).
- **SBERT** is a BERT-based method of using the Siamese structure to derive sentence embeddings that can be compared through cosine similarity (Reimers and Gurevych, 2019).

4.2 Run-time Efficiency

The run-time efficiency is important for sentence representation models because similarity functions are potentially applied to large corpora. In this subsection, we compare *Surrogate*$_{base}$ to InferSent (Conneau et al., 2017), Universal Sent Encoder (Cer et al., 2018), and SBERT$_{base}$ (Reimers and Gurevych, 2019). We adopt a length batching strategy in which sentences are grouped together by length.

The proposed *Surrogate* model is based on PyTorch. InferSent (Conneau et al., 2017) and SBERT (Reimers and Gurevych, 2019) are based on PyTorch. Universal Sent Encoder (Cer et al., 2018) is based on Tensorflow and the model is from the Tensorflow Hub. Model efficiency is measured on a server with Intel i7-5820K CPU @ 3.30GHz, Nvidia Tesla V100 GPU, CUDA 10.2, and cuDNN. We report both CPU and GPU speed and the results can be found in Table 1. As can be seen, InferSent is around 69% faster than *Surrogate* model on CPU since its simpler model architecture. The speed of the proposed *Surrogate* model is comparable to *SBERT* for

Model	CPU	GPU
InferSent	125	1527
Universal Sent Encoder	72	1330
SBERT$_{base}$	41	1315
SBERT$_{base}$ length batching	88	2112
Surrogate$_{base}$	48	1514
Surrogate$_{base}$ length batching	91	2175

Table 1: Computation speed of sentence embedding methods(sentences per second).

both non-batching and batching setups, which is in accord with our expectations due the same transformer structure adopted by the *Surrogate* model.

4.3 Experiment: Semantic Textual Similarity

We evaluate the proposed method on the Semantic Textual Similarity (STS) tasks. We compute the Spearman's rank correlation ρ between the cosine similarity of the sentence pairs and the gold labels for comparison.

Unsupervised Evaluation We evaluate the proposed method on the Semantic Textual Similarity (STS) tasks 2012–2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), the STS benchmark (Cer et al., 2017), and the SICK-Relatedness dataset (Marelli et al., 2014) for evaluation. All datasets contain sentence pairs labeled between 0 and 5 as the semantic relatedness. The proposed models are directly used for inference under the unsupervised setup.

The results are shown in Table 2 and we observe significant performance boosts of the proposed models over baselines. Notably, the proposed models trained in the unsupervised setting (both *Origin* and *Surrogate*) are able to achieve competitive results to models trained on additional annotated NLI datasets. Another observation is, as expected, the *Surrogate* models underperform the *Origin* model as *Origin* serves as an upper bound for *Surrogate* but with a cost of inference speed.

Partially Supervised Evaluation We finetune the model on the combination of the SNLI (Bowman et al., 2015) and the Multi-Genre NLI (Williams et al., 2018) datasets, with the former containing 570K sentence pairs and the latter containing 433K pairs across various genres of sources. Sentence pairs from both datasets are annotated with one of the labels contradiction, entailment, and neutral. The proposed models are trained on the natural language inference task then used for computing sentence representations in an unsupervised manner.

The partially supervised results are shown in Table 2. As can be seen, results from the proposed model finetuned on NLI datasets are comparable to results from unsupervised models since no labeled similarity dataset is used, and comparable to results from supervised models if further finetuned on similarity datasets such as STS.

Supervised Evaluation For the supervised setting, we use the STS benchmark (STSb) to evaluate supervised STS systems. This dataset contains 8,628 sentence pairs from three categories: captions, news, and forums, and is split into 5,749/1,500/1,379 sentence pairs, respectively, for training/dev/test. The proposed models are finetuned on the labeled datasets under the setup.

For our proposed framework, we use *Origin* to represent the original model, where $\boldsymbol{C}$ for each sentence is constructed by searching the entire corpus as in Section 3.3 and we compute similarity scores based on Eq. (4). We also report performances for *Surrogate* models with base and large sizes.

The results are shown in Table 3. We can see that for both model sizes (base and large) and both setups (with and without NLI training), the proposed *Surrogate* model significantly outperforms baseline models, leading to an average of over 2-point performance gains on the STSb dataset.

Note that the *Origin* model cannot be readily adapted to the partially supervised or supervised setting because it is hard to finetune the *Origin* model where the context set $\boldsymbol{C}$ needs to be constructed first. Hence, we finetune the *Surrogate* model as a compensation for the accuracy loss brought by the replacement of *Origin* with *Surrogate*. As we can see from Table 2 and Table 3, finetuning *Surrogate* on NLI datasets and STSb is an effective remedy for the performance loss.

4.4 Experiment: Argument Facet Similarity

We evaluate the proposed model on the Argument Facet Similarity (AFS) dataset (Misra et al., 2016). This dataset contains 6,000 manually annotated argument pairs collected from human conversations on three topics: gun control, gay

Model	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg
				fully unsupervised without human labels				
Avg. Glove embeddings§	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
Avg. Skip-Thought embeddings§	57.11	71.98	61.30	70.13	65.21	59.42	55.50	62.95
InferSent-Glove♯	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Avg. BERT embeddings§	38.78	57.98	57.98	63.15	61.06	46.35	58.40	54.81
BERT [CLS]♯	20.16	30.01	20.09	36.88	38.08	16.50	42.63	29.19
BERTScore♯	54.60	50.11	57.74	70.79	64.58	57.58	51.37	58.11
DPR♯	53.98	56.00	57.83	66.68	67.43	58.53	61.85	60.33
BLEURT♯	70.16	64.97	57.41	72.91	70.01	69.81	58.46	66.25
Universal Sent Encoder♯	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
Origin	**72.41**	**74.30**	**75.45**	**78.45**	**79.93**	**78.47**	**79.49**	**76.93**
$Surrogate_{\text{base}}$	70.62	72.14	72.72	76.34	75.24	74.19	77.20	74.06
$Surrogate_{\text{large}}$	71.93	73.74	73.95	77.01	76.64	75.32	77.84	75.20
				partially supervised without human labels but not the same domain				
InferSent-NLI♯	50.48	67.75	62.15	72.77	66.87	68.03	65.65	64.81
BERT [CLS]-NLI♯	60.35	54.97	64.92	71.49	70.49	73.25	70.79	66.61
BERTScore-NLI♯	60.89	54.64	63.96	74.35	66.67	65.65	66.01	64.60
DPR-NLI♯	61.36	56.71	65.49	71.80	71.03	74.08	70.86	67.33
BLEURT-NLI♯	66.40	68.15	71.98	79.69	77.86	77.98	70.92	73.28
Universal Sent Ecoder-NLI♯	65.55	67.95	71.47	80.81	78.70	78.41	69.31	73.17
$BERT\text{-}NLI^{\sharp}_{\text{base}}$	71.07	76.81	73.29	79.56	74.58	77.10	72.65	75.01
$SBERT\text{-}NLI^{\S}_{\text{base}}$	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.86
$SRoBERTa\text{-}NLI^{\S}_{\text{base}}$	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
$Surrogate\text{-}NLI_{\text{base}}$	74.15	76.50	72.23	81.24	78.75	79.32	78.56	77.25
$BERT\text{-}NLI^{\sharp}_{\text{large}}$	71.62	77.40	72.69	78.61	75.28	77.83	72.64	75.15
$SBERT\text{-}NLI^{\S}_{\text{large}}$	72.27	78.46	74.90	80.99	76.25	79.23	73.75	76.55
$SRoBERTa\text{-}NLI^{\S}_{\text{large}}$	74.53	77.00	73.18	81.85	76.82	79.10	74.29	76.68
$Surrogate\text{-}NLI_{\text{large}}$	**76.98**	**79.83**	**75.15**	**83.54**	**79.32**	**80.82**	**79.64**	**79.33**

Table 2: Spearman rank correlation ρ between the cosine similarity of sentence representations and the gold labels for various Textual Similarity (STS) tasks under the unsupervised setting. We use *-NLI to denote the model additionally trained on NLI datasets. ♯ indicates that results are reproduced by ourselves; § indicates results are taken from Reimers and Gurevych (2019); *Surrogate* are results for our proposed method.

marriage, and death penalty. Each argument pair is labeled on a scale between 0 and 5 with a step of 1. Different from the sentence pairs in STS datasets, the similarity of an argument pair in AFS is measured not only in the claim, but also in the way of reasoning, which makes AFS a more difficult dataset compared to STS datasets. We report the Pearson correlation r and Spearman's rank correlation ρ to compare all models.

Unsupervised Evaluation The results are shown in Table 4, from which we can see that for both the unsupervised settings, the proposed models *Origin* and *Surrogate* outperform baseline models by a large margin, with over 10 points for the unsupervised setting and over 4 points for the supervised setting.

Supervised Evaluation We follow Reimers and Gurevych (2019) to use 10-fold cross-validation for supervised learning. Results are shown in Table 4, from which we can see for both the supervised settings, the proposed models *Origin* and *Surrogate* outperform baseline models by a large margin, with over 10 points for the unsupervised setting and over 4 points for the supervised setting.

4.5 Experiment: Wikipedia Sections Distinction

Ein Dor et al. (2018) constructed a large set of weakly labeled sentence triplets from Wikipedia for evaluating sentence embedding methods, each of which is composed of a pivot sentence, one sentence from the same section, and one from another section. The test set contains 222K triplets. The construction of this dataset is based on the idea that a sentence is thematically closer to sentences within its section than to sentences from other sections.

We use accuracy as the evaluation metric for both unsupervised and supervised experiments: An example is treated as correctly classified if the

Model	Spearman ρ
BERT [CLS]^{sharp}	73.01
BERT$^{\S}_{\text{base}}$	84.30
SBERT$^{\S}_{\text{base}}$	84.67
SRoBERTa$^{\S}_{\text{base}}$	84.92
Surrogate$_{\text{base}}$	**87.91**
BERT-NLI$^{\S}_{\text{base}}$	88.33
SBERT-NLI$^{\S}_{\text{base}}$	85.35
SRoBERTa-NLI$^{\S}_{\text{base}}$	84.79
Surrogate-NLI$_{\text{base}}$	**89.95**
BERT$^{\S}_{\text{large}}$	85.64
SBERT$^{\S}_{\text{large}}$	84.45
SRoBERTa$^{\S}_{\text{large}}$	85.02
Surrogate$_{\text{large}}$	**88.52**
BERT-NLI$^{\S}_{\text{large}}$	88.77
SBERT-NLI$^{\S}_{\text{large}}$	86.10
SRoBERTa-NLI$^{\S}_{\text{large}}$	86.15
Surrogate-NLI$_{\text{large}}$	**90.69**

Table 3: Spearman correlation ρ for the STSb dataset under the supervised setting. We use *-NLI to denote the model additionally trained on NLI datasets. ♯ indicates that results are reproduced by ourselves; § indicates results are taken from Reimers and Gurevych (2019); *Surrogate* are results for our proposed method.

Model	Pearson r	Spearman ρ
	Unsupervised Setting	
Avg. Glove embeddings♯	32.40	34.00
Avg. Skip-Thought embeddings♯	22.34	23.24
InferSent-Glove♯	24.83	25.83
Avg. BERT embeddings♯	29.15	31.45
BERT [CLS]♯	12.00	9.06
BERTScore♯	45.32	33.56
DPR♯	41.89	32.16
BLEURT♯	45.98	44.12
Universal Sent Encoder♯	44.28	43.47
Origin	**56.20**	54.40
Surrogate$_{\text{base}}$	53.00	52.50
Surrogate$_{\text{large}}$	54.50	**54.70**
	Supervised Setting	
BERT [CLS]♯	35.28	36.24
BERT$^{\S}_{\text{base}}$	77.20	74.84
SBERT$^{\S}_{\text{base}}$	76.57	74.13
SRoBERTa$^{\sharp}_{\text{base}}$	77.26	74.89
Surrogate$_{\text{base}}$	79.80	78.20
BERT$^{\S}_{\text{large}}$	78.68	76.38
SBERT$^{\S}_{\text{large}}$	77.85	75.93
SRoBERTa$^{\sharp}_{\text{large}}$	79.03	76.92
Surrogate$_{\text{large}}$	**81.00**	**80.50**

Table 4: Results of Pearson correlation r and Spearman's rank correlation ρ on the Argument Facet Similarity (AFS) dataset. ♯ indicates that results are reproduced by ourselves; § indicates results are taken from Reimers and Gurevych (2019); *Surrogate* are results for our proposed method.

positive example is closer to the anchor than the negative example.

Unsupervised Evaluation We directly evaluate the trained model on the test set without finetuning. Results are shown in Table 5. For the unsupervised setting, the large model *Surrogate*$_{large}$ outperforms the base model *Surrogate*$_{base}$ by 2.1 points.

Supervised Evaluation During training, we use the triple objective to train the proposed model on 1.8M training triplets and evaluate it on the test set.

Results are shown in Table 5. For the supervised setting, the proposed model significantly outperforms SBERT, with a nearly 3-point gain in accuracy for both base and large models.

5 Ablation Studies

We perform comprehensive ablation studies on the STSb dataset with no additional training on NLI datasets to better understand the behavior of the proposed framework. Studies are performed on both the original model setup (denoted by *Origin*) and the surrogate model setup (denoted by *Surrogate*). We adopt the unsupervised setting for comparison.

5.1 Size of Training Data for *Origin*

We would like to understand how the size of data for training *Origin* affects downstream performances. We vary the training size between [10M, 100M, 1B, 10B, 100B] and present the results in Table 6. The model performance drastically improves as we increase the size of training data when its size is below 1B. With more training data, for example, 1B and 10B, the performance approaches the best result achieved with the largest training data.

5.2 Size of C_s

Changing the size of C_s will have an influence on downstream performance. Table 7 shows the results. The overall trend is clear: A larger C leads

Model	Accuracy
Unsupervised Setting	
Avg. Glove embeddings$^\sharp$	60.94
Avg. Skip-Thought embeddings$^\sharp$	61.54
InferSent-Glove$^\sharp$	63.39
Avg. BERT embeddings$^\sharp$	66.40
BERT [CLS]$^\sharp$	32.30
BERTScore$^\sharp$	67.29
DPR$^\sharp$	66.71
BLEURT$^\sharp$	67.39
Universal Sent Encoder$^\sharp$	65.18
$Surrogate_{\text{base}}$	71.40
$Surrogate_{\text{large}}$	73.50
Supervised Setting	
BERT [CLS]$^\sharp$	78.13
$BERT^\sharp_{\text{base}}$	79.30
$SBERT^\S_{\text{base}}$	80.42
$SRoBERTa^\S_{\text{base}}$	79.45
$Surrogate_{\text{base}}$	83.10
$BERT^\sharp_{\text{large}}$	80.15
$SBERT^\S_{\text{large}}$	80.78
$SRoBERTa^\S_{\text{large}}$	79.73
$Surrogate_{\text{large}}$	**83.50**

Table 5: Accuracy results for the Wikipedia sections distinction task. ♯ indicates that results are reproduced by ourselves; § indicates results are taken from Reimers and Gurevych (2019); *Surrogate* are results for our proposed method.

Size	**10M**	**100B**	**1B**	**10B**	**100B**
Spearman ρ	49.41	66.92	76.17	77.81	**78.47**

Table 6: The effect of size of training data for *Origin*.

Size	**20**	**100**	**500**	**1000**
Spearman ρ	66.25	73.93	78.47	**78.56**

Table 7: The effect of size of $\boldsymbol{C}$.

Size	**100K**	**1M**	**10M**	**100M**
Spearman ρ	74.02	76.11	76.92	**77.32**

Table 8: The effect of training data size for *Surrogate*.

to better performance. When the size is 20 or 100, the results are substantially worse than the result when the size is 500. Increasing the size from 500 to 1000 only brings marginal performance gains. We thus use 500 for a trade-off between performance and speed.

Model	Spearman ρ
Full	78.47
w/o discriminative	77.97 (-0.50)
w/o left-context	77.36 (-1.11)
w/o right-context	77.01 (-1.46)
w/o both contexts	76.50 (-1.97)

Table 9: The effect of each term in the scoring function Eq. (3). *discriminative* stands for $\log p(y = 1|\boldsymbol{s}_i, \boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$, *left-context* stands for $\frac{1}{|\boldsymbol{c}_{<i}|} \log p(\boldsymbol{c}_{<i}|\boldsymbol{s}_i, \boldsymbol{c}_{>i})$ and *right-context* stands for $\frac{1}{|\boldsymbol{c}_{>i}|} \log p(\boldsymbol{c}_{>i}|\boldsymbol{c}_{<i}, \boldsymbol{s}_i)$. *both contexts* means we remove both left context and right context.

5.3 Number of Pairs to Train *Surrogate*

Next, we would like to explore the effect of the number of sentence pairs to train *Surrogate*. The results are shown in Table 8. As expected, more training data leads to better performances. With only 100K training pairs, the *Surrogate* model still achieves an acceptable result of 74.02, which indicates that the collected automatically labeled sentence pairs are of high quality.

5.4 How to Construct $\boldsymbol{C}$

We explore the effect of the way we construct $\boldsymbol{C}$. We compare three different strategies: (1) the proposed two-step strategy as detailed in Section 3.3; (2) random selection; and (3) the proposed two-step strategy but without the diversity promotion constraint that allows each text chunk to contribute at most one context. For all strategies, we fix the size of $\boldsymbol{C}$ to 500.

The results for these strategies are, respectively, 78.47, 34.45, and 76.32. The random selection strategy significantly underperforms the other two. The explanation is as follows: Given the huge semantic space for sentences, randomly selected contexts are very likely to be semantic irrelevant to both $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ and can hardly reflect the contextual semantics in which the sentence resides. The similarity computed using context scores based on completely irrelevant contexts is thus extremely noisy, leading to inferior performance. Removing the diversity promotion constraint (the third strategy), the Spearman correlation reduces by over 2 points. The explanation is straightforward: Without the diversity constraint, very similar contexts

Example 1		Score
Sent 1:	the problem likely will mean corrective changes	4.4
	before the shuttle fleet starts flying again .	0.74
Sent 2:	he said the problem needs to be corrected	0.66
	before the space shuttle fleet is cleared to fly again .	0.43
Example 2		**Score**
Sent 1:	every morning, they fly 240 miles to the farm .	0.8
		−0.74
Sent 2:	every morning, you fly 240 miles to every morning .	−0.59
		−0.13
Example 3		**Score**
Sent 1:	rt jones analyst juli niemann said grant was ‘‘the one	1.4
	we were all pulling for he has a very good reputation,’’	−0.71
Sent 2:	rt jones analyst juli niemann said of grant .	−0.39
		0.19

Table 10: We use gold, surrogate, sbert, and universal to denote scores obtained from the gold label, the proposed *Surrogate* model, the SBERT model (Reimers and Gurevych, 2019) and the Universal Sentence Encoder model (Cer et al., 2018), respectively. Scores from the proposed surrogate model are more correlated with the gold compared to the universal sentence encoder and the SBERT model.

will be included in $\boldsymbol{C}$, making the dimensions in the semantic vector redundant; with more diverse contexts, the sentence similarity can be measured more comprehensively and the result can be more accurate.

5.5 Modules in the Scoring Function

We next turn to explore the effect of each term in the scoring function of Eq. (3). Table 9 shows the results. We can observe that removing each of these terms leads to performance drops to different degrees. Removing *discriminative* results in the least performance loss, with a reduction of 0.5; removing *left-context* and *right-context* respectively results in a performance loss of 1.11 and 1.46; and removing both *left-context* and *right-context* has the largest negative impact on the final results, with a performance loss of 1.97. These observations verify the importance of different terms in the scoring function, especially the context prediction terms.

5.6 Model Structures

To train the surrogate model, we originally use the Siamese network structure where two sentences are separately feed into the same model. It would be interesting to see the effect of feeding two sentences together into the model, that is, $\{[\text{CLS}], \boldsymbol{s}_1, [\text{SEP}], \boldsymbol{s}_2\}$ and then using the special token [CLS] for computing the similarity, which is the strategy that BERT uses for sentence-pair classification. Here, we call it the BERT-style model for comparison with the Siamese model.

By training the BERT-style model using the same harvested sentence pairs as the Siamese model with the L_2 regression loss, we obtain a Spearman's rank correlation of 77.43, slightly better than the result of 77.32 for the Siamese model. This is because interactions between words/phrases in two sentences are modeled more sufficiently in the BERT structure as interactions start at the input layer through self-attentions. For the Siamese structure, the two sentences do not interact until the output cosine layer.

The merit of sufficient interactions from the BERT structure also comes at a cost: We need to rerun the full model for any new sentence pair. This is not the case with the Siamese structure, which allows for fast semantic similarity search by caching sentence representations in advance. In practice, we prefer the Siamese structure because the speedup in semantic similarity search overweighs the slight performance boost brought by the BERT structure.

5.7 Case Analysis

We conduct a case analysis on STS benchmark (Cer et al., 2017) test set. Examples can be seen in Table 10. Given two sentences of text s_1 and s_2, the models need to compute how similar s_1 and s_2 are, returning a similarity score between 0 and 5. As can be seen, scores from the proposed *Surrogate* model are more correlated with the gold compared to the universal sentence encoder and the SBERT model.

6 Conclusion

In this work, we propose a new framework for measuring sentence similarity based on the fact that the probabilities of generating two similar sentences based on the same context should be similar. We propose a pipelined system by first harvesting massive amounts of sentence pairs along with their similarity scores, and then training a surrogate model using the automatically labeled sentence pairs for the purpose of faster inference. Extensive experiments demonstrate the effectiveness of the proposed framework against existing sentence embedding based methods.

Acknowledgment

This work is supported by the Science and Technology Innovation 2030 - ‘‘New Generation Artificial Intelligence’’ Major Project (no. 2021ZD0110201) and the Key R & D Projects of the Ministry of Science and Technology (2020YFC0832500). We would like to thank editors for help and anonymous reviewers for their comments and suggestions.

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, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.3115/v1/S14-2010

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics. https://doi.org/10.18653/v1/S16-1081

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. SemEval ’12, pages 385–393, USA. Association for Computational Linguistics.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

Arturs Backurs, Yihe Dong, Piotr Indyk, Ilya Razenshteyn, and Tal Wagner. 2020. Scalable nearest neighbor search for optimal transport. In *International Conference on Machine Learning*, pages 497–506. PMLR.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1075

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Daniel Cer, Yinfei Yang, Sheng-Yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. 2020.

Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*, pages 1542–1553. PMLR.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*. https://doi.org/10.18653/v1/D17-1070

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Liat Ein Dor, Yosi Mass, Alon Halfon, Elad Venezian, Ilya Shnayderman, Ranit Aharonov, and Noam Slonim. 2018. Learning thematic similarity metric from article sections using triplet networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Melbourne, Australia. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-2009

Mamdouh Farouk, Mitsuru Ishizuka, and Danushka Bollegala. 2018. Graph matching based semantic search engine. In *Research Conference on Metadata and Semantics Research*, pages 89–100. Springer. https://doi.org/10.1007/978-3-030-14401-2_8

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*. https://doi.org/10.1080/00437956.1954.11659520

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948, San Diego, California. Association for Computational Linguistics. https://doi.org/10.18653/v1/N16-1108

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California. Association for Computational Linguistics. https://doi.org/10.18653/v1/N16-1162

Gao Huang, Chuan Guo, Matt J. Kusner, Yu Sun, Fei Sha, and Kilian Q. Weinberger. 2016. Supervised word mover's distance. *Advances in Neural Information Processing Systems*, 29:4862–4870.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An easy unsupervised sentence embedding approach. *arXiv preprint arXiv:2104.01767*. https://doi.org/10.18653/v1/2021.findings-emnlp.23

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*. 60:493–502. https://doi.org/10.1108/eb026526

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.550

Seonhoon Kim, Inho Kang, and Nojun Kwak. 2019. Semantic sentence matching with

densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6586–6593. https://doi.org/10.1609/aaai.v33i01.33016586

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2528–2540, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.197

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ryan Kiros, Yukun Zhu, Russ R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems*, 28:3294–3302.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

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

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'shea, and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150. https://doi.org/10.1109/TKDE.2006.130

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.

Hans Peter Luhn. 1957. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317. https://doi.org/10.1147/rd.14.0309

Bill MacCartney and Christopher D. Manning. 2009. *Natural Language Inference*. Citeseer.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.

Amita Misra, Brian Ecker, and Marilyn Walker. 2016. Measuring the similarity of sentential arguments in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 276–287, Los Angeles. Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-3636

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*. https://doi.org/10.18653/v1/D16-1244

Shuang Peng, Hengbin Cui, Niantao Xie, Sujian Li, Jiaxing Zhang, and Xiaolong Li. 2020. Enhanced-rcnn: An efficient method for learning sentence similarity. In *Proceedings of The Web Conference 2020*, WWW '20, pages 2500–2506, New York, NY, USA. Association for Computing Machinery.

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)*, pages 1532–1543. https://doi.org/10.3115/v1/D14-1162

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using Siamese BERT-networks. *arXiv preprint arXiv:1908.10084*. https://doi.org/10.18653/v1/D19-1410

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline M. Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.704

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1179–1189. https://doi.org/10.18653/v1/D17-1122

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Chuanqi Tan, Furu Wei, Wenhui Wang, Weifeng Lv, and Ming Zhou. 2018. Multiway attention networks for modeling sentence pairs. In *IJCAI*, pages 4411–4417.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Bin Wang and C.-C. Jay Kuo. 2020. SBERT-wk: A sentence embedding method by dissecting bert-based word models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2146–2157. https://doi.org/10.1109/TASLP.2020.3008390

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1340–1349, Osaka, Japan. The COLING 2016 Organizing Committee.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans,

Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1101

Lingfei Wu, Ian E. H. Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J. Witbrock. 2018. Word mover's embedding: From word2vec to document embedding. *arXiv preprint arXiv:1811.01713*.

Xinhui Wu and Hui Li. 2017. Topic mover's distance based document classification. In *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pages 1998–2002. IEEE.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

Mingming Yang, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Min Zhang, and Tiejun Zhao. 2019a. Sentence-level agreement for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3076–3082. https://doi.org/10.18653/v1/P19-1296

Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019b. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4699–4709, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1465

Wenpeng Yin and Hinrich Schütze. 2015. MultiGranCNN: An architecture for general matching of text chunks on multiple levels of granularity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 63–73, Beijing, China. Association for Computational Linguistics.

Sho Yokoi, Ryo Takahashi, Reina Akama, Jun Suzuki, and Kentaro Inui. 2020. Word rotator's distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2944–2960, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.236

Mikhail Yurochkin, Sebastian Claici, Edward Chien, Farzaneh Mirzazadeh, and Justin M. Solomon. 2019. Hierarchical optimal transport for document representation. In *Advances in Neural Information Processing Systems*, pages 1601–1611.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTscore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.