Toward Interpretable Semantic Textual Similarity via Optimal Transport-based Contrastive Sentence Learning

Seonghyeon Lee[1]*, Dongha Lee[2], Seongbo Jang[1]*, and Hwanjo Yu[1]†
Computer Science and Engineering, POSTECH, Pohang, Republic of Korea[1]
University of Illinois at Urbana-Champaign (UIUC), Urbana, IL, United States[2]
{sh0416,jang.sb,hwanjoyu}@postech.ac.kr, donghal@illinois.edu

Abstract

Recently, finetuning a pretrained language model to capture the similarity between sentence embeddings has shown the state-of-the-art performance on the semantic textual similarity (STS) task. However, the absence of an interpretation method for the sentence similarity makes it difficult to explain the model output. In this work, we explicitly describe the sentence distance as the weighted sum of contextualized token distances on the basis of a transportation problem, and then present the optimal transport-based distance measure, named RCMD; it identifies and leverages semantically-aligned token pairs. In the end, we propose CLRCMD, a contrastive learning framework that optimizes RCMD of sentence pairs, which enhances the quality of sentence similarity and their interpretation. Extensive experiments demonstrate that our learning framework outperforms other baselines on both STS and interpretable-STS benchmarks, indicating that it computes effective sentence similarity and also provides interpretation consistent with human judgement. The code and checkpoint are publicly available at `https://github.com/sh0416/clrcmd`.

1 Introduction

Predicting the semantic similarity between two sentences has been extensively studied in the literature (Gomaa et al., 2013; Agirre et al., 2015; Majumder et al., 2016; Cer et al., 2017). Several recent studies successfully utilized a pretrained language model such as BERT (Devlin et al., 2019) by finetuning it to capture sentence similarity (Reimers and Gurevych, 2019). To be specific, they define a similarity score between sentence embeddings, which are obtained by aggregating contextualized token embeddings (e.g., avg pooling) or using a special token (e.g., `[CLS]`), then optimize the score for natural language inference (NLI) or semantic textual similarity (STS) tasks (Gao et al., 2021).

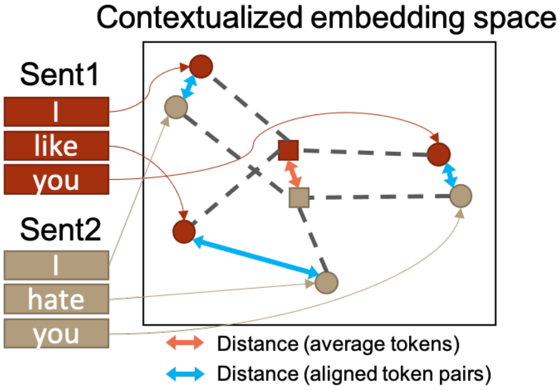

Contextualized embedding space
Sent1
I
like
you
Sent2
I
hate
you
Distance (average tokens)
Distance (aligned token pairs)

Figure 1: An illustrative example of a transportation problem in contextualized embedding space. The existing distance measure between average-pooled sentence embeddings (orange) cannot clearly capture the distances of semantically-aligned token pairs (blue).

Along with the quality of sentence similarity, interpreting the predicted sentence similarity is also important for end-users to better understand the results (Agirre et al., 2016; Gilpin et al., 2018; Rogers et al., 2020). In general, finding out the cross-sentence alignment and the importance of each aligned part is useful for analyzing sentence similarity (Sultan et al., 2015). For example, there were several attempts to use explicit features (e.g., TF-IDF) for easily analyzing the interaction among the shared terms (Salton and Buckley, 1988) or to adopt sophisticated metrics (e.g., word mover's distance) for explicitly describing it by the importance and similarity of word pairs across two sentences (Kusner et al., 2015). However, for recent approaches that leverage sentence embeddings from a pretrained model, it has not been studied how the cross-sentence interaction of each part contributes to the final sentence similarity.

In this work, we propose an analytical method based on optimal transport to analyze existing ap-

*This work was done during internship at Scatterlab.
† Corresponding author

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5969 - 5979
May 22-27, 2022 ©2022 Association for Computational Linguistics

proaches that leverage a pretrained model. We consider a sentence similarity measure a solution to a transportation problem, which aims to transport a collection of contextualized tokens in a sentence to the ones in another sentence. As byproducts of the problem, we obtain a cost matrix and a transportation matrix, which encode the similarities of all token pairs across sentences and their contributions to the sentence similarity, respectively. Using this analytical method, we point out that the existing approaches suffer from the rank-1 constraint in the transportation matrix; this eventually keeps the model from effectively capturing the similarities of semantically-aligned token pairs into sentence similarity. For example, considering transportation in a contextualized embedding space (Figure 1), the distance between averaged token embeddings (orange arrows) cannot clearly represent the distance of semantically-aligned token pairs (blue arrows).

To resolve the above limitation and enhance the interpretability of a model, we present a novel distance measure and a contrastive learning framework that optimizes the distance between sentences. First, we apply optimal transport in a contextualized embedding space and leverage the optimal solution for a relaxed transportation problem as our distance measure. This sentence distance is composed of the distances of semantically-aligned token pairs; this makes the result easily interpretable. Furthermore, we present a contrastive learning framework that adopts the proposed distance to finetune the model with token-level supervision. It optimizes the model to learn the relevance of semantically-aligned token pairs from that of sentence pairs, which further enhances interpretability.

We extensively evaluate our approach and validate the effectiveness of its sentence similarity and interpretation. The comparison on 7 STS benchmarks supports the superiority of sentence similarity predicted by the model trained by our framework. In particular, the evaluation on 2 interpretable-STS datasets demonstrates that the proposed distance measure finds out semantically relevant token pairs that are more consistent with human judgement compared to other baseline methods. Our qualitative analysis shows that both the token alignment and their similarity scores from our model serve as useful resources for end-users to better understand the sentence similarity.

2 Related work

2.1 Semantic textual similarity

Most recent studies tried to leverage a pretrained language model with various model architectures and training objectives for STS tasks, achieving the state-of-the-art performance. In terms of model architecture, Devlin et al. (2019) focus on exhaustive cross-correlation between sentences by taking a concatenated text of two sentences as an input, while Reimers and Gurevych (2019) improve scalability based on a Siamese network and Humeau et al. (2020) adopt a hybrid approach. Along with the progress of model architectures, many advanced objectives for STS tasks were proposed as well. Specifically, Reimers and Gurevych (2019) mainly use the classification objective for an NLI dataset, and Wu et al. (2020) adopt contrastive learning to utilize self-supervision from a large corpus. Yan et al. (2021); Gao et al. (2021) incorporate a parallel corpus such as NLI datasets into their contrastive learning framework.

Despite their effectiveness, the interpretability of the above models for STS tasks was not fully explored (Belinkov and Glass, 2019). One related task is interpretable STS, which aims to predict chunk alignment between two sentences (Agirre et al., 2016). For this task, a variety of supervised approaches were proposed based on neural networks (Konopík et al., 2016; Lopez-Gazpio et al., 2016), linear programming (Tekumalla and Jat, 2016), and pretrained models (Maji et al., 2020). However, these methods cannot predict the similarity between sentences because they focus on finding chunk alignment only. To the best of our knowledge, no previous approaches based on a pretrained model have taken into account both sentence similarity and interpretation.

2.2 Optimal transport

Optimal transport (Monge, 1781) has been successfully applied to many applications in natural language processing (Li et al., 2020; Xu et al., 2021), by the help of its ability to find a plausible correspondence between two objects (Lee et al., 2021a,b). For example, Kusner et al. (2015) adopt optimal transport to measure the distance between two documents with pretrained word vectors. Zhao et al. (2019) adopt optimal transport for evaluating text generation and Zhang et al. (2020) take a greedy approach leveraging pretrained language model. In addition, Swanson et al. (2020) discover the rationale in text-matching via optimal transport, thereby improving model interpretability.

One well-known limitation of optimal transport is that finding the optimal solution is computationally intensive, and thus approximation schemes for this problem have been extensively researched (Grauman and Darrell, 2004; Shirdhonkar and Jacobs, 2008). To get the solution efficiently, Cuturi (2013) provides a regularizer inspired by a probabilistic theory and then uses Sinkhorn's algorithm. Kusner et al. (2015) relax the problem to get the quadratic-time solution by removing one of the constraints, and Wu et al. (2018) introduce a kernel method to approximate the optimal transport.

3 Method

We first analyze the similarity measure used by existing models from the perspective of a transportation problem. Considering the above analysis, we present a novel distance measure and a contrastive sentence learning framework to enhance the interpretability of a sentence similarity model.

3.1 Distance as a transportation problem

We briefly explain the transportation problem and how to interpret the total transportation cost as a distance measure. A transportation problem consists of three components: states before and after transportation, and a cost matrix. In general, the two states are represented in high-dimensional simplex, i.e., $\mathbf{d}^1 \in \Sigma^{d_1}$ and $\mathbf{d}^2 \in \Sigma^{d_2}$, where each dimension implies a specific location with a non-negative quantity. The cost matrix $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$ encodes the unit transportation cost from location i to j into $\mathbf{M}_{i,j}$. In this situation, we search the transportation plan to transport from $\mathbf{d}^1$ to $\mathbf{d}^2$ with the minimum cost. Using the above notations, the optimization problem is written as follows:

$$\underset{\mathbf{T} \in \mathbb{R}_{\geq 0}^{d_1 \times d_2}}{\text{minimize}} \sum_{i,j} \mathbf{T}_{i,j} \mathbf{M}_{i,j} \qquad (1)$$
$$\text{subject to } \mathbf{T}^\top \vec{\mathbf{1}} = \mathbf{d}^2, \mathbf{T}\vec{\mathbf{1}} = \mathbf{d}^1,$$

where each entry of the transportation matrix $\mathbf{T}_{i,j}$ indicates how much quantity is transferred from location i to j. The optimal solution to this problem is called optimal transport, which is also known as earth mover's distance (EMD):

$$d_{\mathbf{M}}^{\text{EMD}} := \sum_{i,j} \mathbf{T}^*_{i,j} \mathbf{M}_{i,j}. \qquad (2)$$

In Equation (2), the distance is computed by the sum of element-wise multiplications of the optimal transportation matrix $\mathbf{T}^*$ and the cost matrix $\mathbf{M}$. In this sense, EMD considers the optimality of distance when combining unit costs in $\mathbf{M}$. That is, the priority of each unit cost when being fused to the distance is encoded in the transportation matrix, which serves as a useful resource for analyzing the distance.

3.1.1 Example: Average pooling

We express cosine similarity with average pooling as a transportation problem and analyze its properties in terms of the transportation matrix. Note that this similarity measure is widely adopted in most of the previous studies (Reimers and Gurevych, 2019; Wu et al., 2020; Gao et al., 2021). Formally, for a sentence of length L, the sentence embedding is generated by applying average pooling to L contextualized token embeddings, i.e., $\mathbf{s} = \frac{1}{L}\sum_{i=1}^{L} \mathbf{x}_i$, where $\mathbf{x}_i$ is the i-th token embedding obtained from a pretrained model. Using the sentence embeddings, the sentence similarity is defined by

$$s^{\text{AVG}} = \cos(\mathbf{s}^1, \mathbf{s}^2) = \frac{\mathbf{s}^{1\top}\mathbf{s}^2}{\|\mathbf{s}^1\|\|\mathbf{s}^2\|}.$$

This average pooling-based sentence similarity can be converted into the distance, $d^{\text{AVG}} = 1 - s^{\text{AVG}}$, described by the token embeddings as follows:

$$d^{\text{AVG}} = 1 - \sum_{i=1}^{L_1}\sum_{j=1}^{L_2} \frac{1}{L_1 L_2} \frac{\|\mathbf{x}_i^1\|\|\mathbf{x}_j^2\|}{\|\mathbf{s}^1\|\,\|\mathbf{s}^2\|} \frac{\mathbf{x}_i^{1\top}\mathbf{x}_j^2}{\|\mathbf{x}_i^1\|\|\mathbf{x}_j^2\|}.$$

From the perspective of Equation (1), this distance is interpreted as a naive solution of a special transportation problem, where the cost matrix and the transportation matrix are

$$\mathbf{M}_{i,j}^{\text{AVG}} = \frac{\|\mathbf{s}^1\|\|\mathbf{s}^2\|}{\|\mathbf{x}_i^1\|\|\mathbf{x}_j^2\|} - \cos(\mathbf{x}_i^1, \mathbf{x}_j^2),$$
$$\mathbf{T}_{i,j}^{\text{AVG}} = \frac{1}{L_1 L_2}\frac{\|\mathbf{x}_i^1\|\|\mathbf{x}_j^2\|}{\|\mathbf{s}^1\|\,\|\mathbf{s}^2\|}. \qquad (3)$$

Each entry of the cost matrix includes negative cosine similarities between token embeddings, and the contribution of each token pair to the sentence distance (i.e., the transportation matrix) is determined by the norms of the token embeddings. In theory, the rank of the transportation matrix is constrained to be one, which prevents effective integration of the token distances into the sentence distance. In practice, it is impossible to involve only semantically-aligned token pairs across sentences because all possible token pairs are considered by the products of their norms. From this analysis, we point out that the average pooling-based similarity is not effective enough to capture the token correspondence between sentences.

3.2 Relaxed optimal transport distance for contextualized token embeddings

To resolve the ineffectiveness of the existing measure, we introduce a novel distance measure based on optimal transport. We first define a transportation problem that considers semantic relevance in a contextualized embedding space. Given the token embeddings of two sentences from a pretrained language model, we construct a cost matrix $\mathbf{M}^{\text{CMD}} \in \mathbb{R}^{L_1 \times L_2}$ that encodes token similarities using cosine distance, and define the state vectors for the two sentences as one vectors normalized by their sentence lengths $\mathbf{d}^1 := \frac{1}{L_1}\vec{\mathbf{1}}$ and $\mathbf{d}^2 := \frac{1}{L_2}\vec{\mathbf{1}}$. As discussed in Section 3.1, we consider the optimal solution to this problem as a distance measure named contextualized token mover's distance (CMD):

$$\begin{aligned} \mathbf{M}^{\text{CMD}}_{i,j} &:= 1 - \cos\left(\mathbf{x}^1_i, \mathbf{x}^2_j\right), \\ d^{\text{CMD}}_{\mathbf{M}} &:= \sum_{i,j} \mathbf{T}^*_{i,j} \mathbf{M}^{\text{CMD}}_{i,j}. \end{aligned}$$

However, finding $\mathbf{T}^*$ incurs huge computational complexity of $O(L^3 \log L)$ where $L = \max(L_1, L_2)$ (Villani, 2008). For this reason, we relax the optimization problem by removing the first constraint, $\mathbf{T}^\top \vec{\mathbf{1}} = \mathbf{d}'$, similar to Kusner et al. (2015). The optimal solution for this relaxed transportation problem is found in $O(L^2)$, keeping the rank of the transportation matrix larger than one. In the end, the optimal transportation matrix and the corresponding distance named relaxed CMD (RCMD) are derived as follows:

$$\begin{aligned} \mathbf{T}^{\text{RCMD}_1}_{i,j} &= \begin{cases} \frac{1}{L_1} & \text{if } j = \operatorname{argmin}_{j'} \mathbf{M}^{\text{CMD}}_{i,j'} \\ 0 & \text{otherwise}, \end{cases} \\ d^{\text{RCMD}_1}_{\mathbf{M}} &:= \frac{1}{L_1} \sum_i \min_j \mathbf{M}^{\text{CMD}}_{i,j}. \end{aligned} \tag{4}$$

Similarly, the elimination of the second constraint, $\mathbf{T}\vec{\mathbf{1}} = \mathbf{d}$, results in $\mathbf{T}^{\text{RCMD}_2}$ and $d^{\text{RCMD}_2}_{\mathbf{M}}$, where the solutions for the two relaxed problems use $\min$ operation on the cost matrix in a row-wise and a column-wise manner, respectively. Note that $\mathbf{T}^{\text{RCMD}_1}$ represents the token-level binary alignment from the first sentence to the second sentence and accordingly the final distance is computed by averaging all the distances of the aligned token pairs. Also, it is obvious that $\mathbf{T}^{\text{RCMD}_1}$ has a much higher rank than $\mathbf{T}^{\text{AVG}}$, which implies that it can express more complex token-level semantic relationship between two sentences.

We remark that our solution provides better interpretability of semantic textual similarity compared to the case of average pooling. For the sentence distance in Equation (3), $\mathbf{T}^{\text{AVG}}$ assigns non-zero values to all token pairs that include irrelevant pairs; this makes it difficult to interpret the result. On the contrary, $\mathbf{T}^{\text{RCMD}_1}$ in Equation (4) is designed to explicitly involve the most relevant token pairs across sentences for the sentence distance, which allows us to interpret the result easily.

3.3 Contrastive sentence similarity learning with semantically-aligned token pairs

We present a contrastive learning framework for RCMD (CLRCMD) that incorporates RCMD into the state-of-the-art contrastive learning framework. To this end, we convert RCMD to the corresponding similarity by $s^{\text{RCMD}_1}_{\mathbf{M}} = 1 - d^{\text{RCMD}_1}_{\mathbf{M}}$:

$$s^{\text{RCMD}_1}_{\mathbf{M}}(\mathbf{s}^1, \mathbf{s}^2) = \frac{1}{L_1} \sum_{i=1}^{L_1} \max_j \cos(\mathbf{x}^1_i, \mathbf{x}^2_j).$$

$s^{\text{RCMD}_2}_{\mathbf{M}}$ is computed in the same manner as well, and we average them to consider bidirectional semantic alignment between two sentences; this provides diverse gradient signals during optimization. The final similarity is described by

$$s^{\text{RCMD}}_{\mathbf{M}}(\mathbf{s}^1, \mathbf{s}^2) := \frac{1}{2}\left(s^{\text{RCMD}_1}_{\mathbf{M}}(\mathbf{s}^1, \mathbf{s}^2) + s^{\text{RCMD}_2}_{\mathbf{M}}(\mathbf{s}^1, \mathbf{s}^2)\right).$$

Adopting this similarity measure, the contrastive learning objective for the i-th sentence pair in a training batch is defined as follows:

$$-\log \frac{\exp(s^{\text{RCMD}}_{\mathbf{M}}(\mathbf{s}^i, \mathbf{s}^i_+)/\tau)}{\sum_{j=1}^{B}(\exp(s^{\text{RCMD}}_{\mathbf{M}}(\mathbf{s}^i, \mathbf{s}^j_+)/\tau) + \exp(s^{\text{RCMD}}_{\mathbf{M}}(\mathbf{s}^i, \mathbf{s}^j_-)/\tau))},$$

where τ is the temperature parameter and B is the batch size. Following (Gao et al., 2021), CLRCMD uses the other sentences in the batch to generate negative pairs.

We argue that CLRCMD enhances both the sentence similarity and its interpretability in the following aspects. First, CLRCMD alleviates the catastrophic forgetting of pretrained semantics during

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg
$\text{BERT}_{\text{base}}$-avg	29.12	59.96	47.22	60.61	63.72	47.20	58.25	52.30
$\text{SBERT}_{\text{base}}$†	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.86
$\text{SBERT}_{\text{base}}$-flow†	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
$\text{SBERT}_{\text{base}}$-whitening†	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
$\text{SimCSE}_{\text{cls}}$-$\text{BERT}_{\text{base}}$†	75.30	84.67	80.19	85.40	80.82	84.25	80.39	81.57
$\text{SimCSE}_{\text{avg}}$-$\text{BERT}_{\text{base}}$	**75.88**	83.28	80.26	86.06	81.33	84.91	79.94	81.67
CLRCMD-$\text{BERT}_{\text{base}}$	75.23	**85.06**	**80.99**	**86.26**	**81.50**	**85.21**	**80.49**	**82.11**
$\text{RoBERTa}_{\text{base}}$-avg	32.50	55.78	45.00	60.61	61.68	55.31	61.66	53.22
$\text{SRoBERTa}_{\text{base}}$†	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
$\text{SRoBERTa}_{\text{base}}$-whitening†	70.46	77.07	74.46	81.64	76.43	79.49	76.65	76.60
$\text{SimCSE}_{\text{cls}}$-$\text{RoBERTa}_{\text{base}}$†	**76.53**	85.21	**80.95**	86.03	82.57	85.83	80.50	82.52
$\text{SimCSE}_{\text{avg}}$-$\text{RoBERTa}_{\text{base}}$	75.75	85.10	80.85	85.95	83.33	85.55	79.41	82.28
CLRCMD-$\text{RoBERTa}_{\text{base}}$	75.68	**85.76**	80.92	**86.58**	**83.48**	**85.89**	**81.01**	**82.76**

Table 1: The results on 7 STS benchmarks. We measure Spearman correlation on all the examples (Gao et al., 2021). † indicates the baseline results reported in their original papers.

the finetuning process. Its token-level supervision is produced by leveraging the textual semantics encoded in a pretrained checkpoint, because token pairs are semantically aligned according to their similarities in the contextualized embedding space. Namely, CLRCMD updates the parameters to improve the quality of sentence similarity while less breaking token-level semantics in the pretrained checkpoint. Furthermore, CLRCMD directly distills the relevance of a sentence pair into the relevance of semantically-aligned token pairs. In this sense, our contextualized embedding space effectively captures the token-level semantic relevance from training sentence pairs, which provides better interpretation for its sentence similarity.

4 Experiments

To analyze our approach in various viewpoints, we design and conduct experiments that focus on the following three research questions:

- **RQ1** Does CLRCMD effectively measure sentence similarities using a pretrained language model?
- **RQ2** Does CLRCMD provide the interpretation of sentence similarity which is well aligned with human judgements?
- **RQ3** Does CLRCMD efficiently compute its sentence similarity for training and inference?

4.1 Training details

We finetune a pretrained model using CLRCMD in the following settings. Following previous work (Gao et al., 2021), we use NLI datasets with hard negatives: SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018). We use a pretrained backbone attached with a single head, which is the same with (Gao et al., 2021). As the initial checkpoint of the pretrained models, we employ `bert-base-uncased` and `roberta-base` provided by huggingface (Devlin et al., 2019; Liu et al., 2019). Adam optimizer is used with the initial learning rate $5e-5$ and linear decay schedule. Fp16 training is enabled where the maximum batch size is 128 on a single V100 GPU, and the softmax temperature is set to $\tau = 0.05$ (Gao et al., 2021). The training is proceeded with 4 different random seeds and the best model is chosen using the best Spearman correlation on STSb validation set which is evaluated every 250 steps during training.

4.2 Semantic textual similarity

We evaluate the similarity model finetuned by CLRCMD for STS task to quantitatively measure the quality of sentence similarity (**RQ1**).

Metric We measure Spearman correlation for each of seven STS benchmarks and calculate their average to compare the capability of representing sentences in general (Conneau and Kiela, 2018).

Baselines We select the baselines that leverage a pretrained model, and they turn out to outperform other traditional baselines. We only list the baseline names for $\text{BERT}_{\text{base}}$ below; the names for $\text{RoBERTa}_{\text{base}}$ are obtained by replacing $\text{BERT}_{\text{base}}$

with RoBERTa$_{\text{base}}$.

- **BERT$_{\text{base}}$-avg** generates sentence embeddings by averaging the token embeddings from BERT$_{\text{base}}$ without finetuning. It indicates zero-shot performance of a checkpoint.
- **SBERT$_{\text{base}}$** (Reimers and Gurevych, 2019) is a pioneering work to finetune a pretrained model for sentence embeddings. It trains a Siamese network using NLI datasets.
- **SimCSE$_{\text{cls}}$-BERT$_{\text{base}}$** (Gao et al., 2021) adopts a contrastive learning framework (Chen et al., 2020) using SNLI and MNLI datasets. The contextualized embedding of `[CLS]` is used as a sentence embedding.
- **SimCSE$_{\text{avg}}$-BERT$_{\text{base}}$** (Gao et al., 2021) is the same with SimCSE$_{\text{cls}}$-BERT$_{\text{base}}$ except that it performs average pooling on token embeddings to obtain a sentence embedding.

Result Table 1 reports Spearman correlation for each dataset and their average. For most of the datasets, CLRCMD shows higher correlation compared to the state-of-the-art baselines. In particular, for STS14, STS15, SICK-R datasets, CLRCMD-BERT$_{\text{base}}$ achieves comparable performance to SimCSE$_{\text{cls}}$-RoBERTa$_{\text{base}}$ whose backbone language model is pretrained with 10 times larger data compared to BERT$_{\text{base}}$. This implies that finetuning with token-level supervision from CLRCMD achieves the performance as good as using an expensively pretrained checkpoint.

4.3 Interpretable semantic textual similarity

Next, we measure the performance of our approach on interpretable STS (iSTS) tasks in order to validate that CLRCMD embeds a sufficient level of interpretability even without any supervision (i.e., labeled training data) about semantically-aligned chunk pairs (**RQ2**).

Experimental setup We utilize the "images" and "headlines" data sources included in SemEval2016 Task 2: iSTS (Agirre et al., 2016). We measure the agreement between human judgement (gold semantic alignment across sentences) and the contributions of all token pairs to sentence similarity (element-wise multiplication of $(1-\mathbf{M})$ and $\mathbf{T}$). One challenge to use our similarity model for this task is to convert token pair contributions into chunk-level alignment. First, we summarize token pair contributions into chunk pair contributions by applying simple average pooling based on the chunk mapping represented by $c(i) = \{k|\text{is\_overlap}(c_i, t_k)\}$, where c_i is the i-th chunk and t_k is the k-th token in a sentence.[1] Then, to obtain the alignment based on the pairwise chunk contributions, we design a criterion for selecting confident chunk pairs (i, j) as follows:

Model	images	headlines
BERT$_{\text{base}}$-avg	82.45	85.98
BERT$_{\text{base}}$-RCMD	**83.00**	**88.25**
SimCSE$_{\text{avg}}$-BERT$_{\text{base}}$	82.98	85.80
CLRCMD-BERT$_{\text{base}}$	**87.25**	**90.55**
RoBERTa$_{\text{base}}$-avg	61.68	52.01
RoBERTa$_{\text{base}}$-RCMD	**82.44**	**88.92**
SimCSE$_{\text{avg}}$-RoBERTa$_{\text{base}}$	73.66	77.30
CLRCMD-RoBERTa$_{\text{base}}$	**84.93**	**88.45**

Table 2: The results on SemEval2016 task 2: iSTS.

$$\mathbf{C}_{i,j} = \frac{1}{|c_1(i)||c_2(j)|}\sum_{k}^{c_1(i)}\sum_{l}^{c_2(j)}\mathbf{T}_{k,l}\mathbf{M}_{k,l},$$

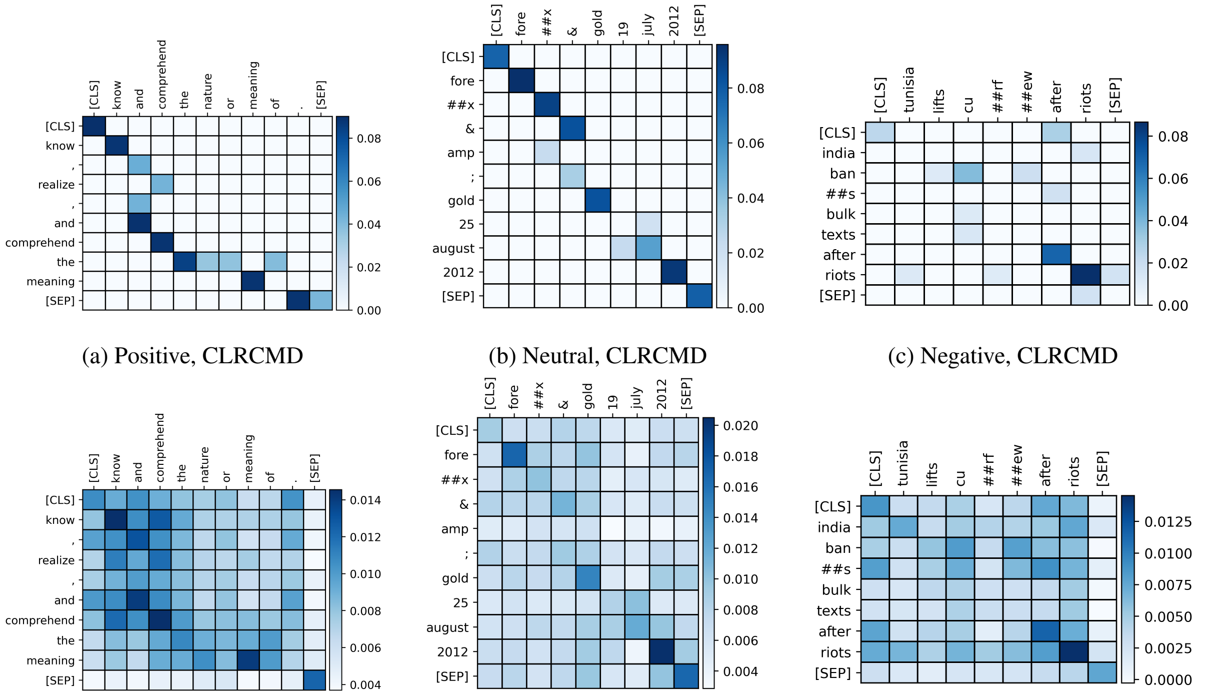
$$a(i,j) = \mathbb{I}[j = \operatorname*{argmax}_{j'}\mathbf{C}_{i,j'}]\cdot\mathbb{I}[i = \operatorname*{argmax}_{i'}\mathbf{C}_{i',j}].$$

Using the aligned chunk pairs obtained by each method, we compute the alignment F1 score as the evaluation metric, which indicates the agreement between human judgement and chunk contribution.[2] We consider eight different configurations to investigate the effectiveness of the following components: 1) sentence similarity, 2) contrastive learning, and 3) pretrained checkpoints.

Result Table 2 shows the clear tendency of iSTS performance with respect to each of the above components. First of all, the token pair contribution from RCMD is more consistent with human judgement than that from average pooling. RCMD improves alignment F1 scores even without finetuning (BERT$_{\text{base}}$-RCMD and RoBERTa$_{\text{base}}$-RCMD), indicating that RCMD effectively discovers the token-level relevance encoded inside a pretrained checkpoint. In addition, the alignment F1 score increases when we finetune a model using CLRCMD. Notably, CLRCMD-BERT$_{\text{base}}$ successfully improves the alignment F1 score whereas

[1] We use gold standard chunking information to focus on alignment only, which is the second subtrack in iSTS.

[2] We employ alignment F1 score implemented in the evaluation script provided by the task organizer.



(a) Positive, CLRCMD

(b) Neutral, CLRCMD

(c) Negative, CLRCMD

(d) Positive, SimCSE$_{\text{avg}}$

(e) Neutral, SimCSE$_{\text{avg}}$

(f) Negative, SimCSE$_{\text{avg}}$

Figure 2: Token pair contribution heatmaps. We use the model finetuned from BERT$_{\text{base}}$ for this experiment.

SimCSE$_{\text{avg}}$-BERT$_{\text{base}}$ does not. This result shows that finetuning a model using the similarity measure based on semantically-aligned token pairs (i.e., fine-grained supervision induced by RCMD) further enhances the interpretability of a model.

4.4 Qualitative analysis

We qualitatively analyze the sentence similarity from the perspective of the transportation problem in order to demonstrate that a model trained by CLRCMD provides clear and accurate explanation (**RQ2**). To this end, we visualize the contribution of token pairs obtained from CLRCMD-BERT$_{\text{base}}$ and that from SimCSE$_{\text{avg}}$-BERT$_{\text{base}}$, and then clarify how their sentence similarity is computed differently from each other. Three sentence pairs are randomly selected from STS13 dataset. Figure 2 illustrates the token pair contribution heatmap for positive, neutral, and negative sentence pairs.

CLRCMD vs. SimCSE$_{\text{avg}}$ Overall, CLRCMD aligns two sentences better than the baseline. To be specific, CLRCMD effectively highlights the contributions of semantically-relevant token pairs and excludes the other contributions (Figure 2 upper). On the contrary, SimCSE$_{\text{avg}}$ fails to represent meaningful token-level relevance for sentence similarity (Figure 2 lower). The rank-1 constraint of SimCSE$_{\text{avg}}$ prevents the model from getting any plausible alignment between two sentences, while it simply tunes the contributions of all possible token pairs at once. We emphasize that the superfluous correlation in the heatmap not only inhibits the capability to capture sentence similarity, but also makes it difficult for humans to understand how sentence similarity is computed.

Case study on positive, neutral, and negative sentence pairs For the positive pair (Figure 2 left), CLRCMD clearly matches all semantically-aligned token pairs including the linking words ({“,”, “and”}–{“and”}), synonyms ({“realize”, “comprehend”}–{“comprehend”}), and omitted contexts ({“the”}–{“the nature of”, “of”}). For the neutral pair (Figure 2 middle), the two sentences have the same lexical structure except for the date. In this case, CLRCMD assigns low contributions to the token pairs about day and month ({“25”, “august”}–{“19”, “july”}), while keeping the contributions high for all the other pairs of identical tokens. Therefore, end-users can clearly figure out which part is semantically different based on their contributions as well as alignment. In case of the negative pair (Figure 2 right), both the models are not able to find any plausible alignment; CLRCMD lowers contributions for most of the token pairs ex-

Batch size	16	32	64	128
RCMD$_{\text{dense}}$	7.5	22.2	OOM	OOM
RCMD$_{\text{sparse}}$	4.6	6.1	10.6	25.8

Table 3: GPU memory usage (GB) of CLRCMD with various batch sizes. OOM: out-of-memory.

cept the token pair with identical contents ("after riots"). That is, end-users also can interpret the negative pair based on the heatmap where semantic correspondence between two sentences does not clearly exist but few overlapped tokens highly contribute to the sentence similarity.

4.5 Resource evaluation

We measure GPU memory usage and inference time of CLRCMD-BERT$_{\text{base}}$ to demonstrate that CLRCMD can be executed on a single GPU and an inference of our model takes almost the same cost to that of the baseline (**RQ3**).

4.5.1 GPU memory usage analysis

Implementation of RCMD We implement two variants of RCMD, RCMD$_{\text{dense}}$ and RCMD$_{\text{sparse}}$, to investigate the effect of exploiting the sparseness in RCMD. Both of them calculate sentence distance by the sum of element-wise multiplications of the cost matrix and the transportation matrix. For an input sentence pair, RCMD$_{\text{dense}}$ maintains the full pairwise token distances ($\mathbf{M}^{\text{CMD}}$), whereas RCMD$_{\text{sparse}}$ only keeps the token distances at which the transportation matrix has nonzero values ($\{\mathbf{M}^{\text{CMD}}_{i,j} | \mathbf{T}^{\text{CMD}}_{i,j} \neq 0\}$). Note that the number of nonzero entries in the transportation matrix of RCMD is at most $2L$, which is an order of magnitude smaller than the number of all entries, L^2.

Result Table 3 reports the GPU memory usage during the finetuning process. For batch-wise contrastive learning, GPU memory requirement becomes $O(B^2)$ in terms of the batch size B, because all pairwise sentence similarities within a batch need to be computed. In this situation, RCMD$_{\text{dense}}$ using a dense matrix drastically increases GPU memory usage by $O(B^2L^2)$, and as a result, the batch size cannot grow larger than 32. In contrast, RCMD$_{\text{sparse}}$ successfully enlarges the batch size up to 128 by exploiting sparseness in the transportation matrix of RCMD, which eventually reduces the space complexity to $O(B^2L)$.

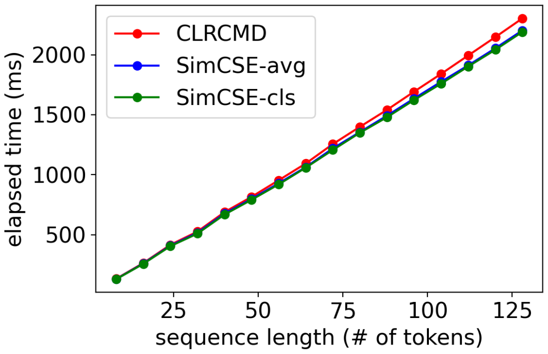


Figure 3: Elapsed time (ms) for the inference of 512 sentence pairs. The result of SimCSE$_{\text{avg}}$ and SimCSE$_{\text{cls}}$ are overlapped.

4.5.2 Inference time analysis

Experimental setup We measure the time for predicting the similarities of 512 sentence pairs on a single V100 GPU while increasing the sequence length from 8 to 128, which is the most influential factor for inference time. We repeat this process 10 times and report the average values.

Result Figure 3 shows the average elapsed time for inference. The model with RCMD takes almost the same inference time as the model with the simple average pooling-based similarity. We highlight that 98% of the sentences in STS13 dataset consist of at most 48 tokens and particularly, the time difference is negligible in case of predicting the sentence pairs whose sentences have less than 48 tokens. This result shows that significant increment of inference time does not occur within the range of the sequence length owing to parallel GPU computations, even though RCMD has the quadratic time complexity with respect to the sentence length.

5 Conclusion

In this work, we present CLRCMD, a learning framework for an interpretable sentence similarity model based on optimal transport. First, we view each sentence similarity measure as a transportation problem, pointing out the unexpressiveness of the existing pooling-based similarity. Integrating the concept of optimal transport into a pretrained language model, CLRCMD defines the distance measure by using the semantically-aligned token pairs between two sentences and furthermore, it finetunes a model with this distance based on contrastive learning for better interpretability. We empirically show that CLRCMD accurately predicts sentence similarity while providing interpretable token pair contributions consistent with human judgements. With the belief that the ability to interpret model behavior is critical for future AI models, we focus on enhancing this virtue targeted on STS task throughout this research.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00584, (SW starlab) Development of Decision Support System Software based on Next-Generation Machine Learning) and the NRF grant funded by the MSIT (South Korea, No.2020R1A2B5B03097210)) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01906, Artificial Intelligence Graduate School Program(POSTECH)).

References

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, Aitor Gonzalez-Agirre, Iñigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uria. 2016. SemEval-2016 task 2: Interpretable semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 512–524, San Diego, California. Association for Computational Linguistics.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

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.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

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. European Language Resources Association (ELRA).

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE.

Wael H Gomaa, Aly A Fahmy, et al. 2013. A survey of text similarity approaches. *international journal of Computer Applications*, 68(13):13–18.

Kristen Grauman and Trevor Darrell. 2004. Fast contour matching using approximate earth mover's distance. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Miloslav Konopík, Ondřej Pražák, David Steinberger, and Tomáš Brychcín. 2016. UWB at SemEval-2016 task 2: Interpretable semantic textual similarity with distributional semantics for chunks. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 803–808, San Diego, California. Association for Computational Linguistics.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR.

Dongha Lee, Su Kim, Seonghyeon Lee, Chanyoung Park, and Hwanjo Yu. 2021a. Learnable structural semantic readout for graph classification. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1180–1185.

Dongha Lee, Seonghyeon Lee, and Hwanjo Yu. 2021b. Learnable dynamic temporal pooling for time series classification. In *AAAI*, pages 8288–8296.

Jianqiao Li, Chunyuan Li, Guoyin Wang, Hao Fu, Yuhchen Lin, Liqun Chen, Yizhe Zhang, Chenyang Tao, Ruiyi Zhang, Wenlin Wang, Dinghan Shen, Qian Yang, and Lawrence Carin. 2020. Improving text generation with student-forcing optimal transport. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9144–9156, Online. Association for Computational Linguistics.

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*.

Iñigo Lopez-Gazpio, Eneko Agirre, and Montse Maritxalar. 2016. iUBC at SemEval-2016 task 2: RNNs and LSTMs for interpretable STS. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 771–776, San Diego, California. Association for Computational Linguistics.

Subhadeep Maji, Rohan Kumar, Manish Bansal, Kalyani Roy, and Pawan Goyal. 2020. Logic constrained pointer networks for interpretable textual similarity. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2405–2411. International Joint Conferences on Artificial Intelligence Organization. Main track.

Goutam Majumder, Partha Pakray, Alexander Gelbukh, and David Pinto. 2016. Semantic textual similarity methods, tools, and applications: A survey. *Computación y Sistemas*, 20(4):647–665.

Gaspard Monge. 1781. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*.

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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Sameer Shirdhonkar and David W Jacobs. 2008. Approximate earth mover's distance in linear time. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2015. DLS@CU: Sentence similarity from word alignment and semantic vector composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 148–153, Denver, Colorado. Association for Computational Linguistics.

Kyle Swanson, Lili Yu, and Tao Lei. 2020. Rationalizing text matching: Learning sparse alignments via optimal transport. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5609–5626, Online. Association for Computational Linguistics.

Lavanya Tekumalla and Sharmistha Jat. 2016. IISCNLP at SemEval-2016 task 2: Interpretable STS with ILP based multiple chunk aligner. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 790–795, San Diego, California. Association for Computational Linguistics.

Cédric Villani. 2008. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media.

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.

Lingfei Wu, Ian En-Hsu 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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4524–4534, Brussels, Belgium. Association for Computational Linguistics.

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*.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary learning via optimal transport for neural machine translation. 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 7361–7373, Online. Association for Computational Linguistics.

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.

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*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.