

Improving Text Generation Evaluation with Batch Centering and Tempered Word Mover Distance

Xi Chen*

Harvard University
Cambridge, MA 02138
chenx@g.harvard.edu

Nan Ding†

Google Research
Venice, CA 90291
{dingnan,

Tomer Levinboim

Google Research
Venice, CA 90291

Radu Soricut

Google Research
Venice, CA 90291

tomerl, rsoricut}@google.com

Abstract

Recent advances in automatic evaluation metrics for text have shown that deep contextualized word representations, such as those generated by BERT encoders, are helpful for designing metrics that correlate well with human judgements. At the same time, it has been argued that contextualized word representations exhibit sub-optimal statistical properties for encoding the true similarity between words or sentences. In this paper, we present two techniques for improving encoding representations for similarity metrics: a batch-mean centering strategy that improves statistical properties; and a computationally efficient tempered Word Mover Distance, for better fusion of the information in the contextualized word representations. We conduct numerical experiments that demonstrate the robustness of our techniques, reporting results over various BERT-backbone learned metrics and achieving state of the art correlation with human ratings on several benchmarks.

1 Introduction

Automatic evaluation metrics play an important role in comparing candidate sentences generated by machines against human references. First-generation metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) use predefined handcrafted rules to measure surface similarity between sentences and have no ability, or very little ability (Banerjee and Lavie, 2005), to go beyond word surface level. To address this problem, later work (Kusner et al., 2015; Zhelezniak et al., 2019) utilize static embedding techniques such as word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) to represent the words in sentences as vectors in a low-dimensional continuous

space, so that word-to-word correlation can be measured by their cosine similarity. However, static embeddings cannot capture the rich syntactic, semantic, and pragmatic aspects of word usage across sentences and paragraphs.

Modern deep learning models based on the Transformer (Vaswani et al., 2017) utilize a multi-layered self-attention structure that encodes not only a global representation of each word (a word embedding), but also its contextualized information within the context considered. Such contextualized word representations have yielded significant improvements on various tasks, including machine translation (Vaswani et al., 2017), NLU tasks (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020), summarization (Zhang et al., 2019a), and automatic evaluation metrics (Reimers and Gurevych, 2019; Zhang et al., 2019b; Zhao et al., 2019; Sellam et al., 2020).

In this paper, we investigate how to better use BERT-based contextualized embeddings in order to arrive at effective evaluation metrics for generated text. We formalize a unified family of text similarity metrics, which operate either at the word/token or sentence level, and show how a number of existing embedding-based similarity metrics belong to this family. In this context, we present a tempered Word Mover Distance (TWMD) formulation by utilizing the Sinkhorn distance (Cuturi, 2013), which adds an entropy regularizer to the objective of WMD (Kusner et al., 2015). Compared to WMD, our TWMD formulation allows for a more efficient optimization using the iterative Sinkhorn algorithm (Cuturi, 2013). Although in theory the Sinkhorn algorithm may require a number of iterations to converge, we find that a single iteration is sufficient and surprisingly effective for TWMD.

Moreover, we follow (Ethayarajh, 2019) and carefully analyze the similarity between contextualized word representations along the different

* Work done during the internship at Google.

† Corresponding author.

layers of a BERT model. We posit three properties that multi-layered contextualized word representations should have (Section 5): (1) zero expected similarity between random words, (2) decreasing out-of-context self-similarity, and (3) increasing in-context similarity between words. As already shown by [Ethayarajh \(2019\)](#), cosine similarity between BERT word-embeddings does not satisfy some of these properties. To address these issues, we design and analyze several centering techniques and find one that satisfies the three properties above. The usefulness of the centering technique and TWMD formulation is validated by our empirical studies over several well-known benchmarks, where we obtain significant numerical improvements and SoTA correlations with human ratings.

2 Related Work

Recent work on learned automatic evaluation metrics leverage pretrained contextualized embeddings by building on top of BERT ([Devlin et al., 2019](#)) or variant ([Liu et al., 2019](#)) representations.

SentenceBERT ([Reimers and Gurevych, 2019](#)) uses cosine similarity of two mean-pooled sentence embedding from the top layer of BERT. BERTscore ([Zhang et al., 2019b](#)) computes the similarity of two sentences as a sum of cosine similarities between maximum-matching tokens embeddings. MoverScore ([Zhao et al., 2019](#)) measures word distance using BERT embeddings and computes the Word Mover Distance (WMD) ([Kusner et al., 2015](#)) from the word distribution of the system text to that of the human reference.

In the next section we propose an abstract framework of embedding-based similarity metrics and show that it contains the metrics mentioned above. We then extend this family of metrics with our own improved evaluation metric.

3 A Family of Similarity Metrics

We consider a family of normalized similarity metrics for both word-level and sentence-level representations parameterized by a function C , as follows:

$$\text{Sim}(\mathbf{x}_1, \mathbf{x}_2) = \frac{C(\mathbf{x}_1, \mathbf{x}_2)}{\sqrt{C(\mathbf{x}_1, \mathbf{x}_1)C(\mathbf{x}_2, \mathbf{x}_2)}}. \quad (1)$$

Clearly, $\text{Sim}(\mathbf{x}, \mathbf{x}) = 1$, and furthermore, if $C(\mathbf{x}_1, \mathbf{x}_2)^2 \leq C(\mathbf{x}_1, \mathbf{x}_1)C(\mathbf{x}_2, \mathbf{x}_2)$, then $\text{Sim}(\mathbf{x}_1, \mathbf{x}_2) \in [-1, 1]$.

For word similarity, \mathbf{x} represents a single word vector. A standard choice is defining $C(\mathbf{x}_1, \mathbf{x}_2) = \langle \mathbf{x}_1, \mathbf{x}_2 \rangle$, the inner product between the two vectors. The resulting word similarity metric $\text{Sim}(\mathbf{x}_1, \mathbf{x}_2) = \left\langle \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|}, \frac{\mathbf{x}_2}{\|\mathbf{x}_2\|} \right\rangle$ becomes the cosine similarity between the two word vectors. If the word vectors are pre-normalized such that $\|\mathbf{x}\| = 1$, then $\text{Sim}(\mathbf{x}_1, \mathbf{x}_2) = \langle \mathbf{x}_1, \mathbf{x}_2 \rangle$.

For sentence similarity, we use $\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^L)$ to denote a $D \times L$ matrix composed by L word vectors belonging to the sentence embedded in a D -dimensional space.

In what follows, we briefly review existing sentence similarity metrics and show that they belong to our family of similarity metrics Eq.(1) with different choices of $C(\mathbf{X}_1, \mathbf{X}_2)$ (with L_1 and L_2 denoting the sentence length for \mathbf{X}_1 and \mathbf{X}_2 , respectively). Note that we do not consider word re-weighting schemes (e.g. by IDF as in ([Zhang et al., 2019b](#))) in this paper, as their contribution does not appear to be consistent over various tasks. In addition, we assume that all word vectors are already pre-normalized.

Sentence-BERT Sentence-BERT ([Reimers and Gurevych, 2019](#)) uses the cosine-similarity between two mean-pooling sentence embeddings. This is the same as Eq.(1) when

$$\begin{aligned} C(\mathbf{X}_1, \mathbf{X}_2) &= \left\langle \frac{1}{L_1} \sum_{i=1}^{L_1} \mathbf{x}_1^i, \frac{1}{L_2} \sum_{j=1}^{L_2} \mathbf{x}_2^j \right\rangle \\ &= \frac{1}{L_1 L_2} \sum_{i=1}^{L_1} \sum_{j=1}^{L_2} \langle \mathbf{x}_1^i, \mathbf{x}_2^j \rangle. \end{aligned}$$

Wordset-CKA Wordset-CKA ([Zhelezniak et al., 2019](#)) uses the centered kernel alignment between the two sentences represented as word sets, where

$$\begin{aligned} C(\mathbf{X}_1, \mathbf{X}_2) &= \text{Tr} \left(\mathbf{X}_1 \mathbf{X}_1^\top \mathbf{X}_2 \mathbf{X}_2^\top \right) \\ &= \sum_{i=1}^{L_1} \sum_{j=1}^{L_2} \langle \mathbf{x}_1^i, \mathbf{x}_2^j \rangle^2. \end{aligned}$$

Here we assume each word embedding \mathbf{x} is pre-centered by the mean of its own dimensions. We refer to this centering method as *dimension-mean centering*.

MoverScore MoverScore ([Zhao et al., 2019](#)) measures the sentence similarity using the Word Mover Distance ([Kusner et al., 2015](#)) from the word

distribution of the hypothesis to that of the gold reference:

$$C(\mathbf{X}_1, \mathbf{X}_2) = \max_{\pi} \sum_{i=1}^{L_1} \sum_{j=1}^{L_2} \pi_{ij} \langle \mathbf{x}_1^i, \mathbf{x}_2^j \rangle$$

$$\text{s.t. } \sum_{i=1}^{L_1} \pi_{ij} = \frac{1}{L_2}, \quad \sum_{j=1}^{L_2} \pi_{ij} = \frac{1}{L_1}. \quad (2)$$

The original MoverScore does not normalize $C(\mathbf{X}_1, \mathbf{X}_2)$ by $\sqrt{C(\mathbf{X}_1, \mathbf{X}_1)C(\mathbf{X}_2, \mathbf{X}_2)}$. In practice, we find the performance to be similar with or without such normalization.

BERTscore BERTscore (Zhang et al., 2019b) introduces three metrics corresponding to recall, precision, and F1 score. We focus the discussion here on BERTscore-Recall, as it performs most consistently across all tasks (see discussions of the precision and F1 scores in Appendix C). BERTscore-Recall uses the sum of cosine similarities between maximum-matching tokens embeddings:

$$C(\mathbf{X}_1, \mathbf{X}_2) = \frac{1}{L_1} \sum_{i=1}^{L_1} \max_{j=1 \dots L_2} \langle \mathbf{x}_1^i, \mathbf{x}_2^j \rangle. \quad (3)$$

For BERTscore, since the words are pre-normalized, we have $C(\mathbf{X}_1, \mathbf{X}_1) = C(\mathbf{X}_2, \mathbf{X}_2) = 1$ and therefore $\text{Sim}(\mathbf{X}_1, \mathbf{X}_2) = C(\mathbf{X}_1, \mathbf{X}_2)$.

Note that BERTscore is closely related to MoverScore, since Eq.(3) is the solution of the Relaxed-WMD (Kusner et al., 2015):

$$C(\mathbf{X}_1, \mathbf{X}_2) = \max_{\pi} \sum_{i=1}^{L_1} \sum_{j=1}^{L_2} \pi_{ij} \langle \mathbf{x}_1^i, \mathbf{x}_2^j \rangle$$

$$\text{s.t. } \sum_{j=1}^{L_2} \pi_{ij} = \frac{1}{L_1}. \quad (4)$$

which is the same as Eq.(2) but without the first constraint.

4 Tempered Word Mover Distance

Word Mover Distance (Kusner et al., 2015) used in MoverScore (Zhao et al., 2019) is rooted in the classical optimal transport distance for probability measures and histograms of features. Despite its excellent performance and intuitive formulation, its computation involves a linear programming solver whose cost scales as $O(L^3 \log L)$ and becomes prohibitive for long sentences or documents with more than a few hundreds of words/tokens.

For this reason, (Kusner et al., 2015) proposed a Relaxed-WMD (RWMD) with only one constraint (see Eq.(4)), which can be evaluated in $O(L^2)$. However, RWMD uses the closest distance without considering there may be multiple words transforming to single words.

Inspired by the Sinkhorn distance (Cuturi, 2013) which smooths the classic optimal transport problem with an entropic regularization term, we introduce the following formulation, which we refer to as tempered-WMD (TWMD):

$$\max_{\pi} \sum_{i=1}^{L_1} \sum_{j=1}^{L_2} \pi_{ij} \langle \mathbf{x}_1^i, \mathbf{x}_2^j \rangle - T \sum_{i=1}^{L_1} \sum_{j=1}^{L_2} \pi_{ij} \log \pi_{ij}$$

$$\text{s.t. } \sum_{i=1}^{L_1} \pi_{ij} = \frac{1}{L_2}, \quad \sum_{j=1}^{L_2} \pi_{ij} = \frac{1}{L_1}. \quad (5)$$

The temperature parameter $T \geq 0$ determines the trade-off between the two terms. When $T = 0$, Eq.(5) reduce to the original WMD as in Eq.(2). When T is larger, (5) encourages more homogeneous distributions.

The added entropy term makes Eq.(5) a strictly concave problem, which can be solved using a matrix scaling algorithm with a linear convergence rate. For example, the Sinkhorn algorithm (Cuturi, 2013) uses the initial condition $\pi_{ij}^0 = \exp\left(-\frac{1}{T} \langle \mathbf{x}_1^i, \mathbf{x}_2^j \rangle\right)$ and alternates between

$$\xi_{ij}^t = \frac{\pi_{ij}^{t-1}}{L_2 \sum_i \pi_{ij}^{t-1}}, \quad \pi_{ij}^t = \frac{\xi_{ij}^t}{L_1 \sum_j \xi_{ij}^t}. \quad (6)$$

The computational cost for each iteration is $O(L^2)$, which is more efficient than to that of WMD. Although in theory this iterative algorithm may require a few of iterations to converge, our experiments show that a single iteration (i.e., $t = 1$) is sufficient and surprisingly effective.

Similarly, a tempered-RWMD (TRWMD) can be obtained by adding an entropy term to Eq.(4):

$$\max_{\pi} \sum_{i=1}^{L_1} \sum_{j=1}^{L_2} \pi_{ij} \langle \mathbf{x}_1^i, \mathbf{x}_2^j \rangle - T \sum_{i=1}^{L_1} \sum_{j=1}^{L_2} \pi_{ij} \log \pi_{ij}$$

$$\text{s.t. } \sum_{j=1}^{L_2} \pi_{ij} = \frac{1}{L_1}.$$

By taking the derivative of the Lagrangian of the above objective, the following closed-form solution is obtained:

$$\pi_{ij}^* = \frac{1}{L_1} \text{softmax}_j \left(\frac{1}{T} \langle \mathbf{x}_1^i, \mathbf{x}_2^j \rangle \right).$$

Plugging in the optimal π_{ij}^* back into the objective yields the following metric:

$$C(\mathbf{X}_1, \mathbf{X}_2) = \frac{T}{L_1} \sum_{i=1}^{L_1} \log \left(\sum_{j=1}^{L_2} \exp \left(\frac{1}{T} \langle \mathbf{x}_1^i, \mathbf{x}_2^j \rangle \right) \right). \quad (7)$$

We note that as $T \rightarrow 0$, $T \log \sum_j \exp(f_j/T) \rightarrow \max_j(f_j)$, and therefore Eq.(7) reduces to Eq.(3).

5 Centered Word Vectors

Ethayarajh (2019) reports that representations obtained by deep models such as BERT exhibit high cosine similarity between any two random words in a corpus, especially at higher layers. They attribute this phenomenon to a highly anisotropic distribution of the word vectors, and further argue that such high similarity represents a bias that blurs the true similarity relationship between word (and sentence) representations and hampers performance in NLP tasks (Mu and Viswanath, 2018). We reproduce here the main results of (Ethayarajh, 2019), including the cosine similarity between two random words (baseline), same words in two different sentences (self-similarity) and two random words in the same sentence (intra-similarity). Figure 1 shows these results for several BERT and BERT-like models. As the leftmost figure shows, most of

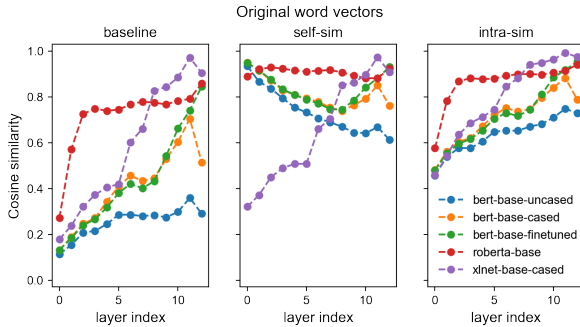


Figure 1: Cosine similarity between two random words (baseline), same words in two different sentences (self-similarity) and two random words in the same sentence (intra-similarity) for five base models, using the original layer representation of words.

these models indeed have a high baseline similarity that quickly increases with layer depth. Ethayarajh (2019) proposes to mitigate this bias by subtracting the baseline similarity from the self-similarity and intra-similarity values (per layer). However, the mathematical and statistical meaning of this solution remains unclear.

In this context, we posit the following three properties that are desirable for word vector representations in context:

1. Zero expected similarity: The word similarity between two random word vectors in the corpus is approximately zero, which indicates random words are unrelated.
2. Decreasing self-similarity: The word similarity between representations of the same word taken from different sentences decreases in higher layers, as each representation encodes more contextual information about its respective sentence.
3. Increasing intra-similarity: The word similarity between different words within the same sentence increases in higher layers, as the words encode more common information about the sentence.

Besides their intuitive appeal, our empirical results (in Section 6) do validate that word representations that obey these properties result in higher performance with respect to modeling similarity.

Since the original word representations does not satisfy these three properties, we explore three methods for centering the word vectors distribution. Consider a corpus \mathcal{C} containing M sentences $\{s_i\}$, each of length N_i . Each word vector is D -dimensional, $\mathbf{w}_{i,j} = [w_{i,j}^{(1)}, \dots, w_{i,j}^{(D)}]$. We propose three candidate word distribution centering approaches:

- Dimension mean centering: centering a word by subtracting the mean of the dimensions within each word vector,

$$\mathbf{v}_{i,j} = \mathbf{w}_{i,j} - \frac{1}{D} \sum_{l=1}^D w_{i,j}^{(l)}.$$

The second term on the RHS is a scalar, which broadcasts to all dimensions of $\mathbf{w}_{i,j}$.

- Sentence mean centering: centering a word by subtracting the mean of the words within the corresponding sentence,

$$\mathbf{v}_{i,j} = \mathbf{w}_{i,j} - \frac{1}{N} \sum_{k=1}^N \mathbf{w}_{i,k}.$$

- Corpus mean centering: centering a word by subtracting the mean of the words in the entire

corpus,

$$\mathbf{v}_{i,j} = \mathbf{w}_{i,j} - \frac{1}{\sum_i N_i} \sum_{i=1}^M \sum_{k=1}^{N_i} \mathbf{w}_{i,k}.$$

We compare these three centering approaches in Figure 2. Due to the layer norm operation in the BERT models, the dimension mean is a small constant that has little effect after subtraction, and therefore it fails on properties 1 and 2 above. The sentence mean centering achieves approximately zero baseline (property 1), but it also reduces the intra-sim to approximately zero (failing property 3). This indicates the subtraction of sentence mean removes the common knowledge of the words about the sentence, which can have a detrimental effect on modeling similarity. Lastly, corpus-mean centering fulfills all three properties above (Fig. 2, bottom row). In this context, we note that, after applying corpus mean centering, cosine-similarity function is reduced to Pearson’s correlation.

Since the computational cost of corpus mean centering can be prohibitive for a large dataset, we consider a batch-mean centering approach, which would be especially useful for fine-tuning tasks. In practice, we find that the values obtained from batch-mean-centered word vectors are very close to those of corpus-mean-centered word vectors. Therefore and henceforth, we use batch-mean centering to approximate the effect of corpus-mean centering.

Finally, it is worth noting that corpus (batch)-mean centering has recently been applied in normalizing multilingual representations (Libovický et al., 2019; Zhao et al., 2020). However, we are the first to demonstrate its superiority over various other centering methods in single-language by analyzing the inter-layer representation similarities.

6 Experiments

In order to demonstrate the effectiveness of our newly proposed approaches, we conduct extensive numerical experiments based on two commonly-used benchmarks: Semantic Textual Similarity (STS), and WMT 17-18 metrics shared task. Our experiments are designed to answer the following questions: (1) Are corpus (batch) centered word vectors better than other centered and un-centered word vectors, across different sentence similarity metrics? (2) How do tempered WMD and RWMD compare to their family-relatives MoverScore and

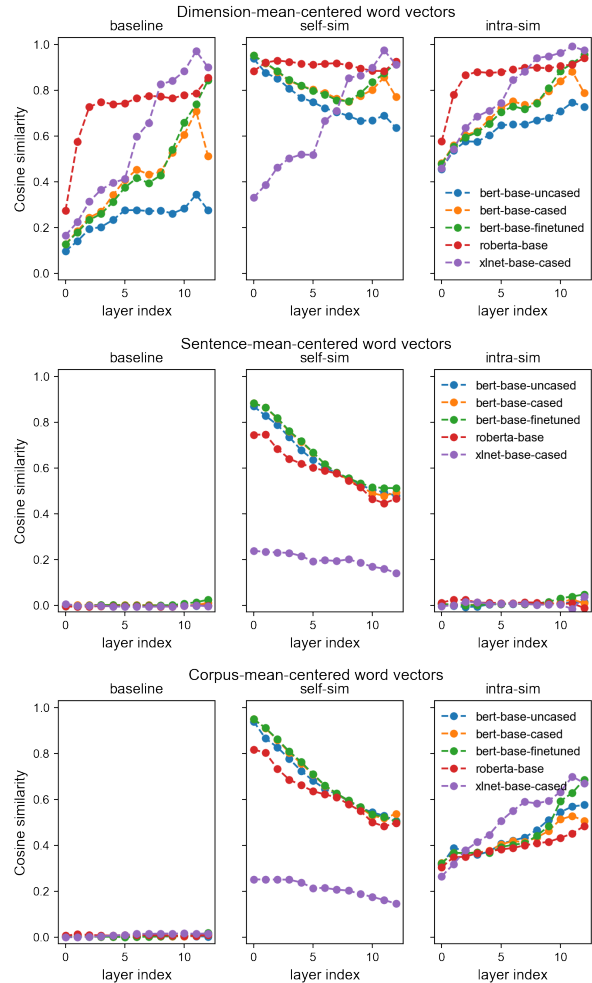


Figure 2: Comparison of the three centering approaches. Dimension mean centering has very little effect. Sentence mean centering removes too much common sentence information. Corpus mean centering shows the correct word contextualization.

BERTscore? (3) How do the temperature hyperparameter and the Sinkhorn iterations affect the performance, and how sensitive are they?

To show that our results are consistent across different BERT variants, we analyze our similarity metrics over four backbone models: bert-base-uncased, bert-large-uncased, roberta-base and roberta-large, all obtained from the Huggingface* Transformers package. (Zhang et al., 2019b) found that the better layers for evaluation metric are usually not the top layer, since the top one is greatly impacted by the pretraining task. In particular, (Zhang et al., 2019b) perform an extensive layer sweep analysis and report that the better layers were always around Layer-10 for the base models, and Layer-19 for the large models. Therefore, in our

*<https://huggingface.co/models>

experiments, we used Layer-10 for all base models, and Layer-19 for all large models. We also present the results of evaluation metrics using different layers in the Appendix A and confirm that our main conclusion is not affected by the choice of layer.

6.1 Semantic Textual Similarity (STS)

The STS benchmark (Agirre et al., 2016) contains sentence pairs and human evaluated scores between 0 and 5 for each pair, with higher scores indicating higher semantic relatedness or similarity for the pair. From 2012 to 2016, it contains 3108, 1500, 3750, 3000, and 1186 records, respectively.

We answer the first question by comparing batch-centered word vectors with other centered and uncentered word vectors using several sentence similarity metrics, including Sentence-BERT, Wordset-CKA, BERTscore and MoverScore.

The results on STS 12-16 for various metrics are shown in Table 1. In general, for all four models (per column: base and large version of BERT and RoBERTa), batch centering gets higher Pearson and Spearman’s correlation of sentence-mean cosine similarity (SBERT) and BERTscore. Dimension-mean centering has very little effect on performance, while sentence-mean improves performance for a few methods. Since Sentence-BERT uses the mean-pooling of the sentence (which would become zero after sentence mean centering), we exclude sentence-mean centering from Sentence-BERT. Overall, batch-mean centering brings an averaged +3.41 / +3.02 improvement, and sentence-mean centering brings an averaged -0.02 / +0.55 on Pearson and Spearman coefficients, across different metrics and models.

6.2 WMT metrics shared task

The WMT metrics shared task is an annual competition for comparing translation metrics against human assessments on machine-translated sentences. We use years 2017 and 2018 of the official WMT test set for evaluation. The 2017 test data includes 3,920 pairs of sentences from the news domain (including a system generated sentence and a groundtruth sentence by human) with human ratings. Similarly, the 2018 test data includes 138,188 pairs of sentences with human ratings but is reported to be much noisier (Sellam et al., 2020).

Evaluation metrics without fine-tuning

We compare the Tempered WMD (TWMD) and TRWMD with the original WMD (Moverscore)

and RWMD (BERTscore) as well as SBERT and WordSet-CKA on WMT 17 and 18. We report the results of RoBERTa-base and RoBERTa-large for WMT 17 and WMT 18, because they appear to be the best performing backbone models for these tasks.

To choose reasonable temperatures for TWMD and TRWMD, we tried a few values between 0.001 and 0.15 on WMT 15-16, and chose for each method based on the best averaged performance (details in Appendix B). The resulting temperatures for TWMD, TRWMD, TWMD-b (where “-b” stands for batch centering of word vectors) and TRWMD-b are $T = 0.02, 0.02, 0.10, 0.15$ respectively. We used a single Sinkhorn iteration for TWMD(-b).

The main results of WMT 17 and 18 are summarized in Table 2 and 3. Batch-mean centering appears to be helpful in improving the scores for all methods. TWMD-b performs the best in most of the cases. In particular, it is on average +2.3 / +2.8 higher than the WMD-based Moverscore-b in WMT-17 and +1.1 / +1.9 higher in WMT-18.

Evaluation metrics with fine-tuning

We also test the effectiveness of batch-mean centering and TWMD in the fine-tuning process. Similar to (Sellam et al., 2020), we make use of the human ratings from WMT 15-16 for training, and evaluate the fine-tuned models on WMT 17 and 18. We use the L2 loss function during fine-tuning,

$$\text{Loss} = \text{MSE}(\text{Sim}(\mathbf{X}_1, \mathbf{X}_2), \hat{y}),$$

where $\mathbf{X}_1, \mathbf{X}_2$ denotes two sentences, and \hat{y} is the human score.

We present the result of TWMD based on the RoBERTa-base and RoBERTa-large backbones in Table 4. We compare the result with that of state-of-the-art BLEURT (Sellam et al., 2020) models. BLEURTbase-pre and BLEURT-pre are directly fine-tuned on WMT 15-16 (with 5344 records in total), while BLEURTbase and BLEURT are additionally pretrained on a large amount of synthetic data from Wikipedia. The scores obtained by TWMD-b not only clearly outperform BLEURTbase-pre and BLEURT-pre with the same training setting, but are comparable or better than the performance of BLEURT with the extra pre-training stage, on both base and large conditions. This last result is especially notable considering that the synthetic data and the task setup used to further pretrain BLEURT were designed with metric

Table 1: Experimental results for various metrics on STS 12-16 datasets (averaged) with BERT/Roberta pretrained checkpoints. The correlations are Pearson (left) and Spearman’s rank (right).

Metric	bert-base-uncased	bert-large-uncased	roberta-base	roberta-large
	r / ρ	r / ρ	r / ρ	r / ρ
SBERT	58.7 / 58.9	56.9 / 57.3	58.0 / 59.6	58.5 / 60.2
SBERT-batch	63.8 / 62.8	62.8 / 62.3	65.9 / 65.1	67.1 / 66.3
SBERT-dim	58.7 / 58.9	56.9 / 57.3	58.0 / 59.6	58.5 / 60.2
CKA	59.8 / 59.5	58.7 / 58.9	58.6 / 59.9	59.1 / 60.4
CKA-batch	60.3 / 61.1	58.9 / 60.0	61.1 / 61.5	62.3 / 62.5
CKA-sent	58.6 / 59.8	59.1 / 60.5	58.7 / 59.2	60.6 / 61.0
CKA-dim	59.8 / 59.5	58.7 / 58.9	58.6 / 59.9	59.1 / 60.4
MoverScore	56.3 / 58.2	54.4 / 56.7	54.8 / 56.2	54.5 / 56.0
MoverScore-batch	58.0 / 60.1	56.2 / 58.6	57.2 / 59.0	57.7 / 59.3
MoverScore-sent	54.2 / 57.4	54.9 / 58.3	54.1 / 56.5	55.9 / 58.1
MoverScore-dim	56.3 / 58.2	54.4 / 56.7	54.8 / 56.2	54.5 / 56.0
BERTscore	59.3 / 59.0	57.7 / 57.8	57.3 / 57.2	57.0 / 57.1
BERTscore-batch	61.1 / 60.9	59.6 / 59.7	60.6 / 60.6	61.5 / 61.4
BERTscore-sent	57.3 / 57.6	58.1 / 58.6	56.8 / 57.2	59.0 / 59.2
BERTscore-dim	59.3 / 59.0	57.7 / 57.8	57.3 / 57.2	57.0 / 57.1

Table 2: Correlation with human scores on the WMT17 Metrics Shared Task. ‘-b’ stands for batch centering of word vectors.

Metric	cs-en	de-en	fi-en	lv-en	ru-en	tr-en	zh-en	Avg.
	τ / r	τ / r	τ / r	τ / r	τ / r	τ / r	τ / r	τ / r
<i>roberta-base</i>								
SBERT	45.1 / 60.0	44.6 / 58.3	58.4 / 69.6	42.9 / 60.6	45.8 / 63.1	46.3 / 52.9	46.0 / 62.0	47.0 / 60.9
SBERT-b	45.2 / 63.4	45.8 / 64.1	56.8 / 74.6	45.1 / 64.9	44.9 / 64.0	47.8 / 63.4	45.4 / 66.1	47.3 / 65.8
CKA	45.0 / 60.5	44.8 / 58.8	58.3 / 70.5	42.8 / 61.0	45.9 / 63.4	46.3 / 53.9	46.1 / 62.4	47.0 / 61.5
CKA-b	48.8 / 68.4	49.1 / 69.1	61.3 / 81.3	48.5 / 69.6	49.6 / 69.6	52.1 / 71.7	49.6 / 70.8	51.3 / 71.5
MoverScore	48.5 / 66.0	47.1 / 65.9	61.6 / 80.9	48.9 / 68.2	51.6 / 69.8	53.8 / 74.2	53.4 / 74.0	52.1 / 71.3
MoverScore-b	47.9 / 66.3	47.3 / 66.1	61.6 / 81.2	48.6 / 68.6	51.4 / 69.8	54.3 / 74.9	52.2 / 72.8	51.9 / 71.3
BERTscore	47.4 / 64.7	48.0 / 66.9	61.9 / 79.9	49.7 / 69.6	50.8 / 69.5	53.4 / 71.3	50.8 / 71.7	51.7 / 70.5
BERTscore-b	47.5 / 66.4	48.8 / 68.7	61.7 / 81.3	49.9 / 70.6	50.7 / 69.8	53.8 / 73.2	49.1 / 70.1	51.6 / 71.5
TWMD	48.3 / 65.8	49.6 / 68.8	62.5 / 81.2	51.3 / 70.5	52.1 / 71.2	54.6 / 73.8	54.7 / 75.5	53.3 / 72.3
TWMD-b	50.0 / 68.5	51.5 / 70.8	63.0 / 82.8	51.9 / 72.3	53.5 / 73.2	56.6 / 77.0	54.0 / 75.0	54.4 / 74.3
TRWMD	47.4 / 64.9	47.9 / 67.0	61.8 / 80.1	49.5 / 69.3	50.9 / 69.5	53.4 / 71.8	50.7 / 71.7	51.7 / 70.7
TRWMD-b	48.5 / 66.8	49.0 / 68.5	61.1 / 81.3	49.5 / 69.3	51.4 / 69.8	54.3 / 74.7	50.2 / 70.8	52.0 / 71.6
<i>roberta-large</i>								
SBERT	50.9 / 67.2	53.1 / 70.8	61.3 / 73.6	51.6 / 70.5	51.4 / 69.0	52.4 / 61.4	51.9 / 68.0	53.2 / 68.6
SBERT-b	47.6 / 66.9	50.7 / 69.5	56.8 / 74.1	47.9 / 67.8	47.3 / 66.4	48.5 / 65.2	47.6 / 67.5	49.5 / 68.2
CKA	51.4 / 68.7	53.4 / 71.3	61.5 / 74.5	51.8 / 71.1	51.8 / 69.3	52.7 / 62.7	52.1 / 68.8	53.5 / 69.5
CKA-b	51.6 / 72.3	54.4 / 74.2	61.8 / 81.6	52.5 / 73.7	53.2 / 73.0	53.6 / 73.8	52.7 / 73.5	54.3 / 74.6
MoverScore	51.6 / 68.8	53.9 / 71.8	62.0 / 81.1	53.4 / 71.7	54.5 / 71.8	56.3 / 76.2	56.3 / 76.1	55.5 / 73.9
MoverScore-b	51.2 / 69.6	53.2 / 71.7	63.1 / 82.1	53.3 / 72.7	54.5 / 72.8	56.8 / 76.9	55.1 / 75.4	55.3 / 74.5
BERTscore	50.9 / 66.9	53.4 / 72.3	61.7 / 79.6	53.5 / 71.6	53.8 / 71.5	54.8 / 71.7	53.9 / 74.4	54.6 / 72.6
BERTscore-b	51.7 / 71.2	53.9 / 74.1	63.6 / 82.5	54.8 / 75.1	54.8 / 73.7	55.6 / 75.0	52.7 / 73.6	55.3 / 75.0
TWMD	52.3 / 69.1	55.7 / 74.4	63.1 / 81.5	54.1 / 72.6	56.0 / 74.1	55.7 / 74.5	57.5 / 77.7	56.3 / 74.9
TWMD-b	53.9 / 73.3	56.4 / 75.9	64.4 / 83.5	55.2 / 75.1	56.9 / 76.2	57.9 / 78.1	56.8 / 77.4	57.4 / 77.1
TRWMD	50.8 / 67.3	53.3 / 72.1	61.5 / 79.7	53.1 / 71.3	54.0 / 71.5	54.5 / 72.0	54.0 / 74.3	54.5 / 72.6
TRWMD-b	52.5 / 71.2	53.9 / 73.4	62.7 / 82.0	53.8 / 73.4	54.8 / 72.8	55.7 / 76.1	53.4 / 74.1	55.3 / 74.7

similarity in mind (by leveraging on classical evaluation metrics for MT such as BLEU and ROUGE), whereas TWMD owes its performance solely to a better use of the representations.

Temperature dependence

The results of TWMD, TRWMD, TWMD-b, TRWMD-b in Table 2 and 3 used the fixed temperature (tuned in WMT15-16) $T = 0.02, 0.02, 0.10, 0.15$ for evaluation. A natural question to ask is how sensitive does the result depend on these hyperparameters.

Figure 3 shows the Pearson correlation vs. temperature for all four models and metrics with different temperature hyperparameters in WMT 15-18. We can see that the TWMD-b and TRWMD-b methods are robust with temperature. In compar-

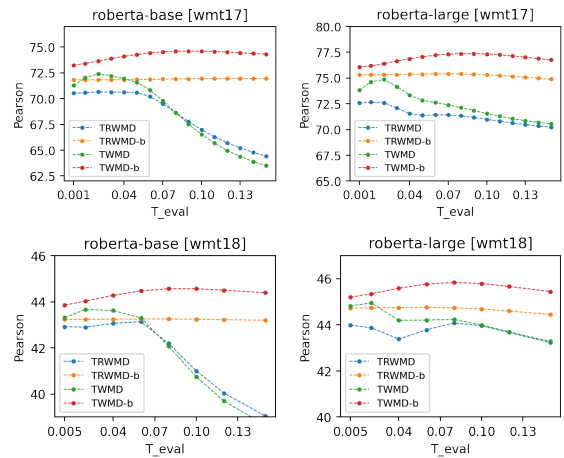


Figure 3: Pearson correlation vs. temperature in evaluation of WMT 17-18.

Table 3: Correlation with human scores on the WMT18 Metrics Shared Task. ‘-b’ stands for batch centering of word vectors.

Metric	cs-en	zh-en	ru-en	fi-en	tr-en	et-en	de-en	Avg.
	τ / r	τ / r	τ / r	τ / r	τ / r	τ / r	τ / r	
<i>roberta-base</i>								
SBERT	26.2 / 34.2	22.9 / 29.0	23.5 / 34.2	23.1 / 32.4	25.3 / 34.6	30.3 / 42.3	35.6 / 48.5	26.7 / 36.5
SBERT-b	28.7 / 40.6	25.8 / 35.6	26.5 / 38.4	25.1 / 36.6	28.7 / 39.8	33.4 / 46.9	38.4 / 54.1	29.5 / 41.6
CKA	26.2 / 34.6	23.0 / 29.5	23.5 / 34.5	23.2 / 32.7	25.4 / 35.1	30.4 / 42.6	35.8 / 48.9	26.8 / 36.9
CKA-b	28.9 / 41.6	26.4 / 37.4	27.2 / 39.9	25.7 / 37.8	30.0 / 42.6	34.2 / 49.3	39.0 / 55.7	30.2 / 43.5
MoverScore	28.5 / 40.6	28.2 / 37.8	27.9 / 39.4	25.1 / 35.8	31.3 / 43.4	34.3 / 48.5	38.9 / 54.2	30.5 / 42.8
MoverScore-b	28.7 / 40.6	28.1 / 37.7	27.8 / 39.2	25.2 / 36.0	31.3 / 43.2	34.4 / 48.6	38.9 / 54.2	30.5 / 42.8
BERTscore	27.6 / 39.5	27.2 / 37.5	27.6 / 39.4	24.9 / 36.0	31.1 / 43.3	34.4 / 48.6	39.3 / 56.1	30.3 / 42.9
BERTscore-b	27.8 / 39.9	26.9 / 37.5	27.5 / 39.3	25.2 / 37.1	31.1 / 43.4	34.4 / 49.0	39.5 / 56.5	30.4 / 43.2
TWMD	28.7 / 40.9	28.2 / 38.4	28.1 / 39.9	25.5 / 36.7	31.7 / 43.9	34.9 / 49.3	39.9 / 56.6	31.0 / 43.7
TWMD-b	29.5 / 42.0	28.4 / 38.9	28.7 / 40.6	26.4 / 38.2	32.3 / 44.5	35.4 / 50.2	40.4 / 57.4	31.6 / 44.6
TRWMD	27.6 / 39.4	27.3 / 37.6	27.6 / 39.3	24.9 / 35.9	31.1 / 43.3	34.4 / 48.5	39.4 / 56.2	30.3 / 42.9
TRWMD-b	28.1 / 39.9	27.6 / 37.6	27.9 / 39.2	25.3 / 36.6	31.4 / 43.2	34.6 / 49.1	39.9 / 56.7	30.7 / 43.2
<i>roberta-large</i>								
SBERT	29.0 / 40.4	24.9 / 33.2	26.6 / 38.2	26.2 / 37.6	28.4 / 39.3	33.9 / 44.1	39.0 / 54.3	29.7 / 41.0
SBERT-b	30.4 / 42.6	26.6 / 35.8	27.9 / 38.9	27.0 / 38.7	29.9 / 41.1	35.0 / 47.1	40.1 / 55.9	31.0 / 42.9
CKA	29.2 / 40.8	25.0 / 33.6	26.7 / 38.6	26.3 / 37.8	28.5 / 39.7	34.0 / 44.7	39.1 / 54.7	29.7 / 41.4
CKA-b	30.4 / 43.9	27.0 / 37.8	28.2 / 41.0	27.0 / 39.5	30.5 / 43.5	35.5 / 50.6	40.4 / 57.6	31.3 / 44.8
MoverScore	29.9 / 41.8	28.7 / 38.0	29.2 / 40.2	26.5 / 37.2	31.9 / 43.6	35.9 / 49.9	40.8 / 56.2	31.9 / 43.8
MoverScore-b	30.0 / 42.1	28.7 / 38.1	28.9 / 40.0	26.5 / 37.5	31.7 / 43.5	35.6 / 49.7	40.4 / 55.6	31.7 / 43.8
BERTscore	29.4 / 41.5	27.9 / 37.9	28.9 / 40.3	26.0 / 36.6	31.6 / 43.6	35.9 / 49.6	41.1 / 58.4	31.6 / 44.0
BERTscore-b	29.7 / 42.6	27.6 / 38.2	28.9 / 40.9	26.4 / 38.4	31.6 / 44.2	35.9 / 50.1	41.0 / 58.5	31.6 / 44.7
TWMD	30.5 / 42.9	28.9 / 39.0	29.5 / 40.9	27.2 / 38.1	32.4 / 44.4	36.5 / 50.5	41.8 / 58.9	32.4 / 45.0
TWMD-b	31.1 / 44.2	28.9 / 39.5	29.7 / 41.8	27.6 / 39.6	32.6 / 45.1	36.7 / 51.3	41.8 / 59.0	32.7 / 45.8
TRWMD	29.2 / 41.4	28.0 / 37.8	28.9 / 40.2	25.9 / 36.5	31.6 / 43.5	35.9 / 49.5	41.1 / 58.3	31.5 / 43.9
TRWMD-b	29.8 / 42.3	28.1 / 38.1	29.2 / 40.6	26.3 / 37.7	31.8 / 43.8	36.0 / 50.2	41.3 / 58.4	31.8 / 44.4

Table 4: Correlation with human scores on the WMT17-18 after fine-tuning on WMT15-16. BLEURTbase and BLEURT have an extra pretraining step, as described in (Sellam et al., 2020)

Metric	WMT-17	WMT-18
	Avg. (τ / r)	Avg. (τ)
<i>base models</i>		
BLEURTbase -pre	56.8 / 75.8	33.6
BLEURTbase	61.0 / 80.2	34.9
TWMDbase	61.7 / 81.0	34.7
TWMDbase-b	61.1 / 80.7	34.7
<i>large models</i>		
BLEURT -pre	59.8 / 79.2	34.5
BLEURT	62.5 / 81.8	35.6
TWMD	62.8 / 81.7	35.5
TWMD-b	63.4 / 82.9	35.5

ision, TWMD and TRWMD without batch-mean centering appears sensitive to the temperature. The Kendall τ correlation follow a similar trend.

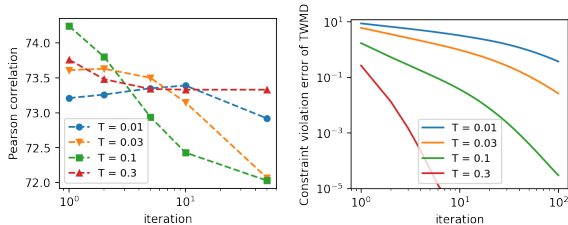


Figure 4: Left: Pearson correlation as a function of the number of iteration in Sinkhorn algorithm. Right: the convergence rate of the Sinkhorn algorithm. The underlying model in this figure is roberta-base with batch mean centered word vectors.

Sinkhorn iteration dependence for Tempered-WMD

We also investigate how the Sinkhorn iterations affect the TWMD-b. (Figure 4, left) shows the Pearson correlation vs the number of Sinkhorn iterations in four different temperatures. Somewhat surprisingly, although Sinkhorn algorithm needs more iterations to converge especially for low temperatures (Figure 4, right), the Pearson correlation of TWMD with only 1 iteration is the highest[†] of the Sinkhorn update for various temperatures.

7 Conclusion

Designing automatic evaluation metrics for text is a challenging task. Recent advances in the field leverage contextualized word representations, which are in turn generated by deep neural network models such as BERT and its variants. We present two techniques for improving such similarity metrics: a batch-mean centering strategy for word representations which addresses the statistical biases within deep contextualized word representations, and a computationally efficient tempered Word Mover Distance. Numerical experiments conducted using representations obtained from a range of BERT-like models confirm that our proposed metric consistently improves the correlation with human judgements.

[†]A minor exception appears to be for $T = 0.01$, where the 1-iter TWMD-b is slightly worse than the 10-iter TWMD-b.

References

- Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511*. ACL (Association for Computational Linguistics).
- Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*.
- Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300.
- 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.
- Kawin Ethayarajh. 2019. How contextual are contextualized word representations? IJCNLP.
- 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.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.
- Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*.
- Chin-Yew Lin. 2004. **Rouge: A package for automatic evaluation of summaries**. In *Text Summarization Branches Out*.
- 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*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeff Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.
- Jiaqi Mu and Pramod Viswanath. 2018. **All-but-the-top: Simple and effective postprocessing for word representations**. In *International Conference on Learning Representations*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of ACL*.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*.
- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. **Bleurt: Learning robust metrics for text generation**.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2019a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.
- Wei Zhao, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein. 2020. Inducing language-agnostic multilingual representations. *arXiv preprint arXiv:2008.09112*.
- 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. *arXiv preprint arXiv:1909.02622*.
- Vitalii Zhelezniak, April Shen, Daniel Busbridge, Aleksandar Savkov, and Nils Hammerla. 2019. Correlations between word vector sets. *arXiv preprint arXiv:1910.02902*.