Moving Other Way: Exploring Word Mover Distance Extensions*

Ilya S. Smirnov and Ivan P. Yamshchikov[a]

LEYA Lab, Yandex, Higher School of Economics, Russia

Keywords: Semantic Similarity, WMD, Word Mover's Distance, Hyperbolic Space, Poincare Embeddings, Alpha Embeddings.

Abstract: The word mover's distance (WMD) is a popular semantic similarity metric for two documents. This metric is quite interpretable and reflects the similarity well, but some aspects can be improved. This position paper studies several possible extensions of WMD. We introduce some regularizations of WMD based on a word match and the frequency of words in the corpus as a weighting factor. Besides, we calculate WMD in word vector spaces with non-Euclidean geometry and compare it with the metric in Euclidean space. We validate possible extensions of WMD on six document classification datasets. Some proposed extensions show better results in terms of the k-nearest neighbor classification error than WMD.

1 INTRODUCTION

Semantic similarity metrics are essential for several Natural Language Processing (NLP) tasks. When working with paraphrasing, style transfer, topic modeling, and other NLP tasks, one usually has to estimate how close the meanings of two texts are to each other. The most straightforward approaches to measure the distance between documents rely on some scoring procedure for the overlapping bag of words (BOW) and term frequency-inverse document frequency (TF-IDF). However, such methods do not incorporate any semantic information about the words that comprise the text. Hence, these methods will evaluate documents with different but semantically similar words as entirely different texts. There are plenty of other metrics of semantic similarity: chrF (Popović, 2015) - character n-gram score that measures the number of n-grams that coincide both input and output; BLEU (Papineni et al., 2002) developed for automatic evaluation of machine translation, or the BERT-score proposed in (Zhang et al., 2019) for estimation of text generation. In (Yamshchikov et al., 2020) authors show that many current metrics of semantic similarity have significant flaws and do not entirely reflect human evaluations. At the same time, these evaluations themselves are pretty noisy and heavily depend on the crowd-sourcing procedure from (Solomon et al., 2021).

One of the most successful metrics in this regard is Word Mover's Distance (WMD) (Kusner et al., 2015). WMD represents text documents as a weighted point cloud of embedded words. It specifies the geometry of words in space using pretrained word embeddings and determines the distances between documents as the optimal transport distance between them. Many NLP tasks, such as document classification, topic modeling, or text style transfer, use WMD as a metric to automatically evaluate semantic similarity since it is reliable and easy to implement. It is also relatively cheap computationally and has an intuitive interpretation. For these reasons, this paper experiments with possible extensions of WMD. In this position paper, we discuss possible ways to improve WMD without losing its interpretability and without making the calculation too resource-intensive.

We perform a series of experiments calculating WMD for different pretrained embeddings in vector spaces with different geometry. The list includes Hyperbolic space (Dhingra et al., 2018) and tangent space of the probability simplex represented with Poincare embeddings (Nickel and Kiela, 2017; Tifrea et al., 2018) and Alpha embeddings (Volpi and Malagò, 2021). We suggest several word features that might affect WMD performance. We also discuss which directions seem to be the most promising for further metric improvements.

[a] https://orcid.org/0000-0003-3784-0671

*This work is an output of a research project implemented as part of the Basic Research Program at the National Research University Higher School of Economics (HSE University).

Smirnov, I. and Yamshchikov, I.
Moving Other Way: Exploring Word Mover Distance Extensions.
DOI: 10.5220/0011096900003197
In *Proceedings of the 7th International Conference on Complexity, Future Information Systems and Risk (COMPLEXIS 2022)*, pages 92-97
ISBN: 978-989-758-565-4; ISSN: 2184-5034

2 WORD MOVER'S DISTANCE

Word Mover's Distance is fundamentally based on the Kantorovich problem between discrete measures, which is one of the fundamental problems of optimal transport (OT). Formally speaking, there should be several inputs to calculate the metric:

- two discrete measures or probability distributions α, β:

  $$\alpha = \sum_{i=1}^{n} \mathbf{a}_i \delta_{x_i}, \beta = \sum_{i=1}^{m} \mathbf{b}_i \delta_{y_i}$$

  where $x_1, \cdots, x_n \in R^d$, $y_1, \cdots, y_m \in R^d$, δ_x is the Dirac at position x, **a** and **b** are Dirac's weights. One also has a contract that $\sum_{i=1}^{n} \mathbf{a}_i = \sum_{i=1}^{m} \mathbf{b}_i$, and she tries to move the first measure to the second one so that they are equal.
- The transportation cost matrix **C**:

  $$\mathbf{C}_{ij} = c(x_i, y_j)$$

  where $c(x_i, y_j) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a distance or transportation cost between x_i and y_j

Earth Mover's Distance or solution of the Kantorovich problem between α and β is then defined through the following optimization problem:

$$EMD(\alpha, \beta, \mathbf{C}) = \min_{P \in \mathbb{R}} \sum_{i,j} \mathbf{C}_{ij} \mathbf{P}_{ij}$$

$$s.t. \quad \mathbf{P}_{ij} \geq 0,\ \mathbf{P}\mathbb{1} = \mathbf{a},\ \mathbf{P}^T \mathbb{1} = \mathbf{b}$$

$\mathbf{P}_{ij}$ intuitively represents the "amount" of word i transported to word j.

Vanilla WMD is the cost of transporting a set of word vectors from the first document, represented as a bag of words, into a set of word vectors from the second document in a Euclidean space. So it is just an EMD but with some conditions:

- probability distribution in terms of a document:

  $$\mathbf{a} = \sum_{i=1}^{n} a_i \delta_{w_i}$$

  $$s.t. \quad \sum_{i=1}^{n} a_i = 1$$

  where $w_1, \cdots, w_n$ is the set of words in a document, a_i stands for the number of times the word w_i appeared in the document divided by a total number of words in a document.
- $\mathbf{C}(w_i, w_j) = \|w_i - w_j\|^2$

Now that we have described WMD in detail let us discuss several essential results that emerged since the original paper (Kusner et al., 2015), where it was introduced for the first time.

3 RELATED WORK

Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) are the most famous semantic embeddings of words based on their context and frequency of co-occurrence in text corpora. They leverage the so-called distributional hypothesis (Harris, 1954), which states that similar words tend to appear in similar contexts. Word2Vec and Glove vectors are shown to effectively capture semantic similarity at the word level. Word Mover's Distance takes this underlying word geometry into account but also utilizes the ideas of optimal transport and thus inherits specific theoretical properties from OT. It is continuously used and optimized for various tasks.

(Huang et al., 2016) propose an efficient technique to learn a supervised WMD via leveraging semantic differences between individual words discovered during supervised training. (Yokoi et al., 2020) demonstrate in their paper that Euclidean distance is not appropriate as a distance metric between word embeddings and use cosine similarity instead. They also weight documents' BOW with L2 norms of word embeddings. (Wang et al., 2020) replace assumption that documents' BOWs have the same measure to solve Kantorovich problem of optimal transport with the usage of Wasserstein-Fisher-Rao distance between documents based on unbalanced optimal transport principles. The work of (Sato et al., 2021) is especially significant for further discussion. The authors re-evaluate the performances of WMD and the classical baselines and find that once the data gets L1 or L2 normalization, the performance of other classical semantic similarity measures becomes comparable with WMD. The authors also show that WMD performs better with TF-IDF regularization. In high-dimensional spaces, WMD behaves similarly to BOW, while in low-dimensional spaces, it seems to be influenced by the dimensionality curse.

(Sun et al., 2018) show that WMD performs quite well on a hierarchical multilevel structure.

4 EXPERIMENTAL SETTINGS

This section describes the experiments that we carry out in detail.

4.1 Datasets

To assure better reproducibility, we work with the datasets presented in (Kusner et al., 2015) and (Sato

Table 1: Information about used datasets.

	twitter	imdb	amazon	classic	bbcsport	ohsumed
Total number of texts	3115	1250	1500	2000	737	1250
Train number of texts	2492	1000	1200	1600	589	1000
Test number of texts	623	250	300	400	148	250
Average word's L2 norm	2.60	2.76	2.77	2.86	2.78	3.03
Average text's length in words	10.57	87.12	165.50	54.50	144.79	91.55

Table 2: kNN classification errors on all datasets for some variations of WMD. The results are reported on the best number of neighbors selected from $[1;20]$.

	twitter	imdb	amazon	classic	bbcsport	ohsumed
WMD	29.4 ± 1.7	24.8	9.7 ± 2.3	5.7 ± 1.9	3.4 ± 1.7	92.4
WMD-TF-IDF	29.2 ± 1.0	**21.2**	9.0 ± 1.7	4.9 ± 1.5	2.7 ± 1.0	92.0
WRD	**28.7 ± 1.3**	23.2	**7.2 ± 2.1**	**4.1 ± 1.3**	3.6 ± 1.1	90
OPT_1	29.6 ± 1.2	27.2	20.7 ± 4.0	15.2 ± 12.9	4.1 ± 1.5	**88.5**
OPT_2	29.6 ± 1.5	27.2	10.1 ± 1.5	5.8 ± 1.8	**2.6 ± 1.1**	92.0

et al., 2021)[1]. For the evaluation, we use six datasets that we believe to be diverse and illustrative enough for the aims of this discussion. The datasets are TWITTER, IMDB, AMAZON, CLASSIC, BBC SPORT, and OHSUMED.

We remove stop words from all datasets, except TWITTER, as in the original paper (Kusner et al., 2015). IMDB and OHSUMED datasets have predefined train/test splits. On four other datasets, we use 5-fold cross-validation. Due to time constraints to speed up the computations, we take subsamples from more extensive datasets. Table 1 shows the parameters of all the resulting datasets we use for the evaluation and experiments.

Similar to (Sato et al., 2021), we split the training set into an 80/20 train/validation set and select the neighborhood size from [1, 20] using the validation dataset.

4.2 Embeddings

Since WMD relies on some form of word embeddings, we experiment with several pre-trained models and train several others ourselves. First, we use original 300-dimensional Word2Vec embeddings trained on the Google News corpus that contains about 100 billion words[2]. A series of works hint that original Euclidian geometry might be suboptimal for the space of word embeddings.

(Nickel and Kiela, 2017; Tifrea et al., 2018) suggest the Poincare embeddings that map words in a hyperbolic rather than a Euclidian space. Hyperbolic space is a non-Euclidean geometric space with an alternative axiom instead of Euclid's parallel postulate. In hyperbolic space, circle circumference and disc area grow exponentially with radius, but in Euclidean space, they grow linearly and quadratically, respectively. This property makes hyperbolic spaces particularly efficient to embed hierarchical structures like trees, where the number of nodes grows exponentially with depth. The preferable way to model Hyperbolic space is the Poincare unit ball, so all embeddings v will have $\|v\| \leq 1$. Poincare embeddings are learned using a loss function that minimizes the hyperbolic distance between embeddings of similar words and maximizes the hyperbolic distance between embeddings of different words.

(Volpi and Malagò, 2021) proposes alpha embeddings as a generalization to the Riemannian case where the computation of the cosine product between two tangent vectors is used to estimate semantic similarity. According to Information Geometry, a statistical model can be modeled as a Riemannian manifold with the Fisher information matrix and a family of α connections. Authors propose a conditional Skip-Gram model that represents an exponential family in the simplex, parameterized by two matrices U and V of size $n \times d$, where n is the cardinality of the dictionary, and d is the size of the embeddings. Columns of V determine the sufficient statistics of the model, while each row u_w of U identifies a probability distribution. The alpha embeddings are defined up to the choice of a reference distribution p_0.The natural alpha embedding of a given word w is defined as the projection of the logarithmic map onto the tangent space of some submodel. We will not dive into more detail since this is beyond the scope of our work but address the reader to the original paper (Volpi and Malagò, 2021) .

We train Poincare embeddings for Word2Vec and

[1] The data is available online https://github.com/mkusner/wmd

[2] https://code.google.com/archive/p/word2vec/

alpha embeddings over GloVe in 8 different dimensionalities on text8 corpus[3] that contains around 17 million words. The SkipGram models for both Word2Vec and Poincare embeddings are trained with similar parameters: the minimum number of occurrences of a word in the corpus is 50, the size of the context window is 8, negative sampling with 20 samples. The number of epochs is 5, and the learning rate decreases from 0.025 to 0.

Similar to (Volpi and Malagò, 2021) we use GloVe embeddings as a base for alpha embeddings and train it for fifteen epochs. The word2vec Skip-Gram with negative sampling is equivalent to a matrix factorization with GloVe so it is easy to reproduce using the original framework[4]. The co-occurrence matrix is built with the minimum number of occurrences of a word in the corpus being 50 and the window size equal to 8.

4.3 Models

We use only vanilla WMD to compare embedding in different spaces, but the distance between word embeddings is measured differently depending on the geometry of the underlying space. We set hyperparameter α in tangent space of the probability simplex equal to 1 to ease distance computation.

- Euclidean space

  $$c(w_i, w_j) = \|w_i - w_j\|^2$$

- Hyperbolic space (Unit Poincare ball)

  $$c(w_i, w_j) = \cosh^{-1}\left(1 + 2\frac{\|w_i - w_j\|^2}{(1 - \|w_i\|^2)(1 - \|w_j\|^2)}\right)$$

- Tangent space of the probability simplex

  $$c(w_i, w_j) = \frac{w_i^T I(p_u) w_j}{\|w_i\|_{I(p_u)} \|w_j\|_{I(p_u)}}$$

  where $I(p_u)$ is the Fisher information matrix which could be computed during training alpha embeddings

To compare WMD variations on pretrained word embeddings, we also compare five variations of WMD. The main idea of the WMD variants that we experiment with is that one wants to prioritize the transportation of rare words. Naturally, the semantics of a rare word might carry far more meaning than the several frequently used ones. According to (Arefyev et al., 2018) the embedding norm of a word positively correlates with its frequency in the training corpus. We use this idea and propose the following WMD variations for comparison:

- vanilla WMD
- WMD with TF-IDF regularization applied to bags of words for both documents (Sato et al., 2021)
- WRD - Word Rotator's Distance (Yokoi et al., 2020)

  to compute it authors use

  $$1 - cos(w_i, w_j)$$

  as a distance or transportation cost between words w_i and w_j. They multiply the document's BOW by words norm. More precisely, let a document have N unique words and $A = a_1, \cdots, a_N$, where a_i is a number of times a word w_i occurs in the document. New BOW is calculated like this:

  $$A' = a_1 \cdot \|w_1\|, \cdots, a_n \cdot \|w_n\|$$

- OPT_1: after calculating vanilla WMD between two documents, which have BOWs named as A and B, we normalize the WMD score by the following coefficient:

  $$coeff = 1 + \sum_{w_a = w_b} \frac{min(a, b)}{\|w_a\|^2}$$

  where w_a and a are a word and its frequency in the first document respectively, while w_b and b stand for the word and its frequency in the second one . This coefficient makes WMD lower if there are rare matching words in both documents.

- OPT_2: we want to increase the measure of rare words relative to the rest of the words. So let's use rebalanced BOW with the formula inspired by TF-IDF:

  $$A' = a_1 \cdot \log\left(\frac{d}{\|w_1\|}\right), \cdots, a_n \cdot \log\left(\frac{d}{\|w_n\|}\right)$$

  where d is the dimensionality of word embeddings.

  There is a simple idea behind this: the norm of rare words is less than that of the frequently used ones, $\log\left(\frac{d}{\|w\|}\right)$. Thus we increase the impact of rare words more while decreasing the effects of the frequent ones less.

4.4 Evaluation and Results

Table 2 shows that overall our variations of WMD could behave quite badly. OPT_1 with a simple division of the final metric by a coefficient is especially

[3] https://deepai.org/dataset/text8

[4] https://github.com/rist-ro/argo

Table 3: kNN classification errors on TWITTER dataset for all embeddings' types and different embeddings' dimensions. The best number of neighbors was selected from the segment $[1;20]$.

	Word2Vec embeddings	Poincare embeddings	Alpha embeddings
5	**33.1 ± 2.7**	35.0 ± 3.7	36.4 ± 4.3
10	**33.2 ± 3.3**	36.7 ± 5.3	36.5 ± 4.7
25	**33.8 ± 2.8**	34.7 ± 3.5	35.4 ± 4.5
50	**33.5 ± 2.5**	37.6 ± 6.4	37.1 ± 6.5
100	34.4 ± 3.2	**33.2 ± 2.9**	38.2 ± 6.7
200	**34.4 ± 3.2**	34.7 ± 3.4	35.5 ± 4.0
300	34.5 ± 3.4	36.6 ± 4.2	**33.1 ± 2.4**
400	34.9 ± 3.8	34.4 ± 3.6	**34.4 ± 1.9**

Table 4: kNN classification errors on IMDB dataset for all embeddings' types and different embeddings' dimensions. The best number of neighbors was selected from the segment $[1;20]$.

	5	10	25	50	100	200	300	400
Word2Vec embeddings	43	**35**	**29**	**30**	**24**	**30**	**28**	**28**
Poincare embeddings	**39**	43	42	50	45	45	41	41
Alpha embeddings	52	48	63	49	52	48	57	55

crude when comparing it on all datasets. However, on some of the tasks, the proposed measures are either the best or close to the best result.

So OPT_1 shows the best performance on the OHSUMED dataset, which contains medical abstracts categorized by different disease groups. This dataset has an abundance of rare words, thus it seems that the proposed normalization was useful because of this property of the data. Its bad performance on other datasets could be due to an excessive amount of frequent word matches in those documents.

Looking at Tables 3 and 4, one can notice that on both datasets WMD with Word2Vec embeddings performs well and beats WMD with other embeddings. However, there are some outliers. On the TWITTER dataset, alpha embeddings perform best for the standard standard dimension of 300, which may signal the possible benefits of further studying them and learning or iterating over the hyperparameter α.

For embeddings of dimension 5 on the IMDB dataset, Poincare embeddings perform the best. Thus one could suggest that they capture semantics in low-dimensional spaces better than other embeddings' types.

We can also notice that on TWITTER the classification error is almost the same, whereas on IMDB the differences are noticeable. It seems that Poincare and Alpha embeddings better reflect the semantics of frequently used words.

5 DISCUSSION

We want to point out some interesting moments for future research.

Geometry of the Underlying Space. The Euclidean embedding space must be of large dimension. Other geometries show better results at lower dimensions. However, we are experimenting with small samples of two datasets. It would be interesting to check whether the superiority trend of Word2Vec embeddings in terms of WMD continues on embeddings of large dimensions for other datasets or larger subsamples of TWITTER and IMDB datasets.

Normalization with Word Frequencies. The frequency of words in the training corpus affects the WMD score, but we make only several attempts to use it. This seems to be a promising direction for future research. Indeed, on the specialized datasets the variants that take word frequencies into account show good results.

6 CONCLUSIONS

This position paper conducts a series of experiments to calculate Word Mover's Distance in different embedding spaces.

It seems that taking into account the frequency of words and improving the mechanism of optimal transport in application to semantics could be promising directions for further research. However, additional work on this problem is necessary.

Further, new embedding types have been found that behave well on specific dimensions, and further study of these embeddings can be meaningful within the framework of the semantic similarity problem.

REFERENCES

Arefyev, N., Ermolaev, P., and Panchenko, A. (2018). How much does a word weigh? weighting word embeddings for word sense induction. *arXiv preprint arXiv:1805.09209*.

Dhingra, B., Shallue, C. J., Norouzi, M., Dai, A. M., and Dahl, G. E. (2018). Embedding text in hyperbolic spaces. *arXiv preprint arXiv:1806.04313*.

Harris, Z. (1954). Distributional hypothesis. *Word*, 10(23):146–162.

Huang, G., Quo, C., Kusner, M. J., Sun, Y., Weinberger, K. Q., and Sha, F. (2016). Supervised word mover's distance. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4869–4877.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30:6338–6347.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

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

Popović, M. (2015). chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Sato, R., Yamada, M., and Kashima, H. (2021). Reevaluating word mover's distance. *arXiv preprint arXiv:2105.14403*.

Solomon, S., Cohn, A., Rosenblum, H., Hershkovitz, C., and Yamshchikov, I. P. (2021). Rethinking crowd sourcing for semantic similarity. *arXiv preprint arXiv:2109.11969*.

Sun, C., Ng, K. T. J., Henville, P., and Marchant, R. (2018). Hierarchical word mover distance for collaboration recommender system. In *Australasian Conference on Data Mining*, pages 289–302. Springer.

Tifrea, A., Becigneul, G., and Ganea, O.-E. (2018). Poincare glove: Hyperbolic word embeddings. In *International Conference on Learning Representations*.

Volpi, R. and Malagò, L. (2021). Natural alpha embeddings. *Information Geometry*, pages 1–27.

Wang, Z., Zhou, D., Yang, M., Zhang, Y., Rao, C., and Wu, H. (2020). Robust document distance with wasserstein-fisher-rao metric. In *Asian Conference on Machine Learning*, pages 721–736. PMLR.

Yamshchikov, I. P., Shibaev, V., Khlebnikov, N., and Tikhonov, A. (2020). Style-transfer and paraphrase: Looking for a sensible semantic similarity metric. *arXiv preprint arXiv:2004.05001*.

Yokoi, S., Takahashi, R., Akama, R., Suzuki, J., and Inui, K. (2020). Word rotator's distance. *arXiv preprint arXiv:2004.15003*.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.