Boosting Tricks for Word Mover's Distance

Konstantinos Skianis, Fragkiskos D. Malliaros, Nikolaos Tziortziotis, Michalis Vazirgiannis

Boosting Tricks for Word Mover's Distance

Konstantinos Skianis[1], Fragkiskos D. Malliaros[2], Nikolaos Tziortziotis[3], and Michalis Vazirgiannis[4]

[1] BLUAI, Athens, Greece
skianis.konstantinos@gmail.com
[2] Paris-Saclay University, CentraleSupélec, Inria, France
fragkiskos.malliaros@centralesupelec.fr
[3] Jellyfish, France
ntziorzi@gmail.com
[4] École Polytechnique, Palaiseau, France
mvazirg@lix.polytechnique.fr

Abstract. Word embeddings have opened a new path in creating novel approaches for addressing traditional problems in the natural language processing (NLP) domain. However, using word embeddings to compare text documents remains a relatively unexplored topic — with Word Mover's Distance (WMD) being the prominent tool used so far. In this paper, we present a variety of tools that can further improve the computation of distances between documents based on WMD. We demonstrate that, alternative stopwords, cross document-topic comparison, deep contextualized word vectors and convex metric learning, constitute powerful tools that can boost WMD.

Keywords: Word mover's distance · Word embeddings · Text classification.

1 Introduction

Measuring distance between documents has always been a key component in many natural language processing tasks, such as document classification [2], machine translation [38], question answering [3] and text generation [5]. Nevertheless, the task can present various difficulties, making it not trivial; whether two documents are similar or not, is not always clear and may vary from application to application.

Following a naive, but effective in many cases, assumption, previous similarity measures that make use of the vector space model [29], were treating words in a document as if they were independent to each other. On the contrary, the distributional hypothesis [13], stated that words that co-occur in similar contexts and frequently, tend to have similar meanings and share common semantics.

With the rise of neural networks and deep learning methods in the natural language processing community [1, 6], word embeddings [22] have had a huge impact in numerous tasks. Apart from constituting the most popular input for

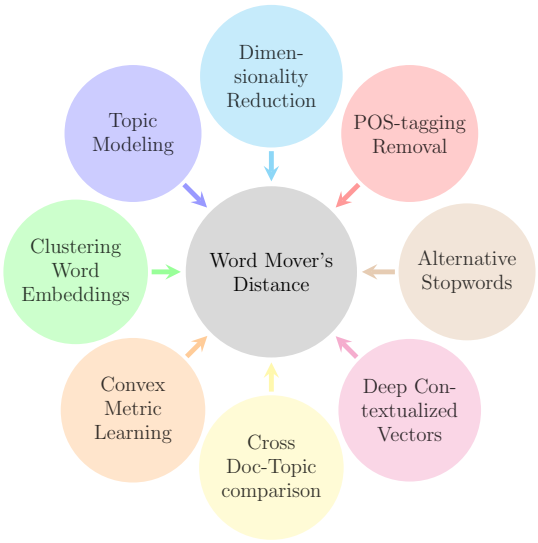

Fig. 1. Areas and tools that could be utilized to boost Word Mover's Distance.

CNNs [18] and LSTMs [15], word embeddings have been used to compute similarity between documents that might not carry any identical words.

Succeeding the idea of using Earth Mover's Distance to measure document distance [33], Kusner et al. [19] presented *Word Mover's Distance* (WMD), a method for measuring the dissimilarity between two text documents as the minimum amount of distance that the embedded words of one document need to travel to reach the embedded words of another document. Moving forward, a supervised version of Word Mover's Distance has been introduced [14], which employs metric learning techniques when label knowledge exists. Their approaches have shown unprecedented results in the task of text classification via k nearest neighbor ($k$nn).

Although Word Mover's Distance is a powerful method for comparing two text documents, it can fall into the case where a word is very common and thus not contributing to measuring the distance. Moreover the exact computation of WMD scales at $\mathcal{O}(n^3)$, making it prohibitive for large collections of documents with big vocabularies. Kusner et al. [19] addressed this problem with a much faster variant, the *Relaxed WMD*, which is a lower bound of the exact WMD.

As WMD consists of multiple components, several improvement suggestions can be done. We observe that many tools can be of service, such as: a) dimensionality reduction, where some of the dimensions are actually useful, or dropping the number of dimensions may help as well making the computation faster; b) POS-tagging removal, where we care mainly about nouns and verbs; c) testing alternative stopwords and which stopwords to remove; d) topic modelling, adding words that belong in the same topic; e) clustering word embeddings; f)

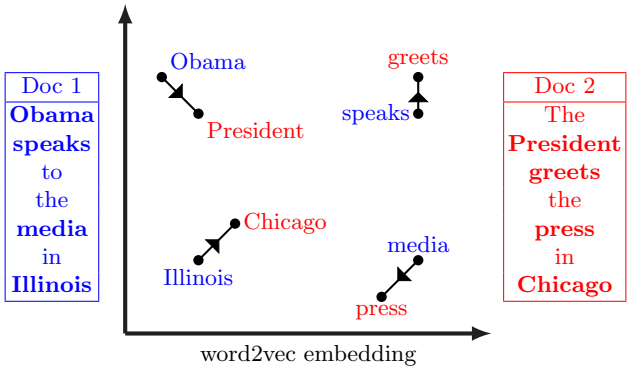

Fig. 2. An illustration of the Word Mover's Distance by [19]. The distance between the two documents is given by the minimum cumulative distance that all the words in Doc 1 need to travel in order to reach the words of Doc 2.

cross doc-topic comparison, adding neighbour words; g) contextualized vectors, like the recently introduced ELMo [24]; h) metric learning, assuming we want to add label information. An illustration of the components is shown in Fig. 1.

In this work, we have focused on testing alternative stopwords, cross document-topic comparison, deep contextualized word vectors and convex metric learning, by examining how they can further improve the performance of text categorization based on the WMD. Our approach is summarized as follows.

- First, by selecting specific *stopwords*, we observe that they play a significant role in the distance computation process.
- Next, utilizing *cross document-topic comparison*, we aim to make the comparison of two documents more meaningful by employing additional neighbour words.
- Finally, in order to boost the supervised version of WMD (S-WMD), we apply two state-of-the-art convex metric learning algorithms, namely *Large Margin Nearest Neighbors* (LMNN) [35], as well as *Maximally Collapsing Metric Learning* (MCML) [11].

Roadmap. In Section 2 we introduce the background and related work needed for the rest of paper. Next, in Section 3 we present our focused contribution on boosting Word Mover's Distance and Supervised Word Mover's Distance. Our experiments and results follow in Section 4. Finally, in Section 5 we conclude our study and present future work directions.

2 Background and Related Work

Let's assume that we have access to a word2vec [22] embedding matrix $X \in \mathbf{R}^{n \times m}$ for a finite size vocabulary of n words. $x_i \in \mathbf{R}^m$ represents the embedding of the i-th word in a m-dimensional space.

Word Mover's Distance Word Mover's Distance tries to embody the semantic similarity between individual word pairs into the document distance metric. Let d and d' be the nBOW representation of two documents, and $T \in R^{n \times n}$ be a flow matrix where $T_{ij} \geq 0$ denotes how much of word i in d travels to word j in d'. More precisely, the distance between word i and word j becomes $c(i,j) = ||x_i - x_j||_2$. By $c(i,j)$ we point to the cost associated with "traveling" from one word to another. To transform d entirely into d' we ensure that the entire outgoing flow from word i equals d_i, i.e. $\sum_j T_{ij} = d_i$. Formally, the minimum cumulative cost of moving d to d' given the constraints is provided by the solution to the following linear program:

$$\begin{aligned} \text{minimize} \quad & \sum_{i,j=1}^{n} T_{ij} c(i,j) \\ \text{subject to:} \quad & \sum_{j=1}^{n} T_{ij} = d_i \quad \forall i \in \{1,\dots,n\} \\ & \sum_{i=1}^{n} T_{ij} = d'_j \quad \forall j \in \{1,\dots,n\}, \end{aligned} \tag{1}$$

where $T_{ij} >= 0$ denotes how much of word i in d travels to word j in d'.

In Figure 2 we present a schematic illustration of the Word Mover's Distance, between two documents "Obama speaks to the media in Illinois" and "The President greets the press in Chicago".

Relaxed word moving distance (RWMD). Although WMD is powerful, it comes with a high complexity. Thus the authors in [19] relaxed the WMD optimization problem by removing one of the two constraints. If just the second constraint is removed, the optimization becomes:

$$\begin{aligned} \text{minimize} \quad & \sum_{i,j=1}^{n} T_{ij} c(i,j) \\ \text{subject to:} \quad & \sum_{j=1}^{n} T_{ij} = d_i \quad \forall i \in \{1,\dots,n\}. \end{aligned} \tag{2}$$

RWMD, which can be seen as an approximation of WMD, is much faster, making it more efficient for large documents.

Topics in Word Embeddings In terms of utilizing topics, Das et al. [7] introduced Gaussian Latent Dirichlet Allocation, a method for topic modeling on word embeddings, treating the document as a collection of word embeddings and topics itself as multivariate Gaussian distributions in the embedding space. Later, the authors of [23] presented a novel document similarity measure based on the definition of a graph kernel between two pairs of documents. By representing

each document as a graph-of-words, various approaches were able to model these relationships and then determine how similar two documents are by using a modified shortest-path graph kernel [27, 21]. Skianis et al. [31] clusters the word vectors and extracts topics from the embedding space. The work by Kim et al. [17] utilizes Word Mover's Distance to identify related words when no direct matches are found between a query and a document. In recent work, a topical distance approach [36] was attempted using word embeddings, by iteratively picking words from a vocabulary that closes the topical gap between documents.

List	#	Description
nltk (3.2.2)	153	Van Rijsbergen (1979) [34] and Porter (1980) [25]
spaCy (2.0.9)	305	Improved list from [32] extra words: former, beside, done, whither, sometimes
Gensim (3.7.1)	337	Same as spaCy (Improved list from [32] extra words: thick, computer, cry, system, bill
SMART	571	SMART (System for the Mechanical Analysis and Retrieval of Text) Information Retrieval System developed at Cornell University in the 1960s.
ROUGE	598	Extended SMART list used in ROUGE 1.5.5 Summary Evaluation Toolkit extra words: reuters, ap, news, tech, index, 3 letter days of the week and months.
Terrier	733	Terrier Retrieval Engine
ATIRE	988	Puurula (2013) [26]

Table 1. A sample of the most popular stopword lists available. The first three are integrated in well-known Python NLP libraries (versions of tools are mentioned inside parenthesis).

Metric Learning Metric or distance learning is a field covering both supervised and unsupervised techniques [37]. As an extension of Word Mover's Distance, Supervised Word Mover's Distance [14] was presented, a method which utilized Neighborhood Component Analysis (NCA) [12] along with word embeddings and documents labels. While S-WMD is powerful, its loss function is nonconvex and is thus highly dependent on the initial setting of A and w.

Apart from NCA, there exists a plethora of popular methods for generalized Euclidean metric learning. Information-Theoretic Metric Learning (ITML) [8] learns a metric by minimizing a KL-divergence subject to generalized Euclidean distance constraints.

3 Boosting WMD and S-WMD

Our work is focused on studying the contribution of the following three tools in the computation of WMD: vocabulary trimming with stopword removal, cross document-topic comparison and convex metric learning methods. In the next paragraphs, we present in detail each of those tools.

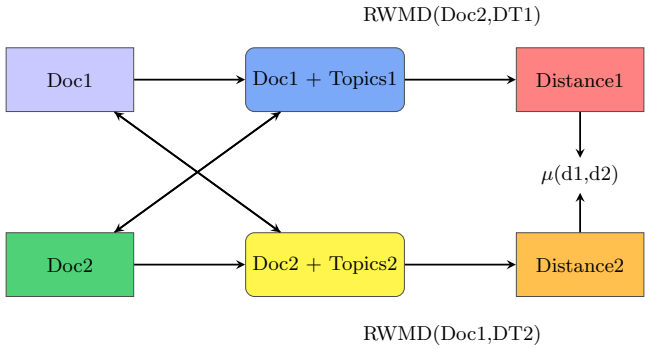

Fig. 3. Cross document-topic comparison schematic. With Topics we refer to neighbor or centroid words. DT1 stands for adding Doc1 words with Topics1.

3.1 Alternative Stopwords

In the natural language processing domain, stop words are generally the most common words in a language. For plenty of natural language processing tasks, these words are normally filtered out, making the vocabulary's size of the text set to be analyzed smaller. More specifically stopword removal is advised for text classification (or categorization) and caption generation, but not for tasks like machine translation, text summarization and language modeling.

Vocabulary pruning can help us to get rid of insignificant words, making the Relaxed WMD faster, while producing a better comparison between documents. In this way, we can make the "travel cost" needed from one document to another cheaper, faster and more effective, as the remaining words are the actual words that contribute to the meaning. Stopwords are in general category independent and thus the first that one could consider irrelevant. In recent work [30], stopword removal has been studied especially for topic modelling.

Nevertheless, there is no single universal list of stopwords used by all natural language processing tools, and indeed not all tools even use such a list. In the Word Mover's Distance paper, Kusner et al. [19] used the stopword list provided by the SMART system [28], composed of 571 words. [32] introduced another stopword list for English, with 339 words. Later, Puurula [26] created a new stopword list with 988 words (ATIRE). Along the study of this paper, we found more than 10 different stopword lists that are currently being used across many NLP tasks. Our goal is to examine how a different stopword list can affect the distance computation.

In Table 1 we present a sample of the most popular stopword lists used in the NLP community, as well as some integrated in well-known Python libraries.

3.2 Cross Document-Topic Comparison

Words that compose documents are sometimes not adequate to indicate the topics covered. Following this intuition, we augment the word space of each

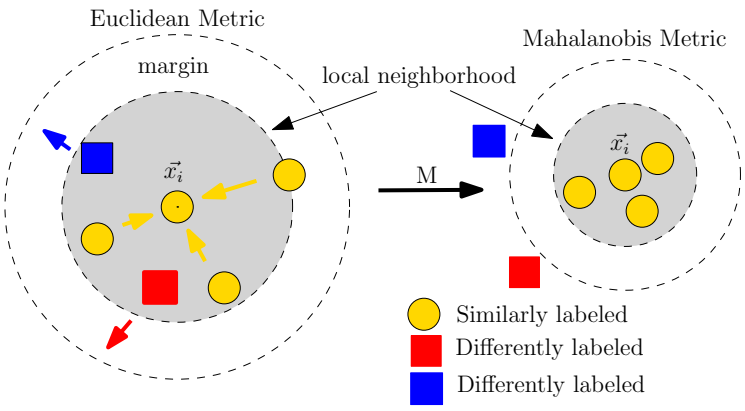

Fig. 4. An illustration of the LMNN algorithm [35]. In text, documents of the same class are pushed together in order to be closer and those of different class to be further away.

document by adding neighbors of each word. That way, the documents become more descriptive and carry more specific information. Our initial approach is to apply $k$nn search for each word in a document. Then, we either add these words' vectors, or create a centroid of the word's neighbors, adding it as a "topic-word".

Nevertheless, looking for the nearest neighbors of each word inside the global word vector space for every document is expensive. Thus, we apply clustering in the word vector space beforehand, and then search for the nearest neighbors of each word in the topic that the word belongs to. Here we introduce the concept of "topic-words", which we refer to either neighbors or centroids of a word's neighbors. In our settings, we have used hard k-means clustering. After extracting these "topic-words", we cross compare with the Relaxed Word Mover's Distance approach. Finally, we quantify the distance as the mean of the previous two RWMD distances. Our proposed scheme is shown in Fig. 3.

3.3 Deep Contextualized Word Representations

Recent work by [24] introduced ELMo, a novel type of deep contextualized word representations. These vectors represent internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus. In their work, the vectors used are derived from a bidirectional LSTM that is trained with a coupled language model (LM) objective. Essentially, for every word there is a vector every time it is found around a context.

In our work, we replace Google's pretrained vectors with ELMo, to test how it can affect measuring distances. This is expected, as ELMo is proved to boost many diverse NLP tasks, even with a simple averaging without any fine-tuning. To the best of our knowledge, our work is the first to incorporate deep contextualized word representations as an input for distance computation.

3.4 Convex Metric Learning

We extend our work in supervised settings, similarly to Supervised WMD [14]. Here, we propose to replace the Neighborhood Component Analysis method, which includes a non-convex cost function [12] with convex ones. These are the Maximally Collapsing Metric Learning (MCML) [11] and Large Margin Nearest Neighbors (LMNN) [35].

Both methods carry the property of learning a metric where points in the same class are simultaneously near each other and far from points in the other classes. As the $k$nn rule relies heavily on the underlying metric (a test input is classified by a majority vote amongst its k nearest neighbors), it is a good indicator for the quality of the metric in use. We present a schematic illustration of LMNN in Fig. 4. Similarly labeled text documents are pushed together in order to be closer and those differently labeled tend to be further away.

Maximally Collapsing Metric Learning (MCML). Maximally Collapsing Metric Learning (MCML) [11] was introduced as a linear learning algorithm for quadratic Gaussian metrics (Mahalanobis distances) used in supervised classification tasks. The method is based on the simple geometric intuition that a good metric is one under which points in the same class are simultaneously near each other and far from points in the other classes. A convex optimization problem is formulated, whose solution generates such a metric by trying to collapse all instances within the same class to a single point, while pushing other class instances infinitely far away.

Large Margin Nearest Neighbors (LMNN). Later, Large Margin Nearest Neighbors (LMNN) [35] came along, a metric that encourages inputs with similar labels to be close in a local region, and inputs of different labels to be farther by a large margin. LMNN is an algorithm to learn a Mahalanobis metric specifically to improve the error of $k$nn classification.

4 Experiments

Datasets. We evaluate in the context of $k$nn classification on six document categorization tasks:

1. Bbcsport: articles between 2004-2005;
2. Twitter: set of tweets labeled with sentiments 'positive', 'negative', or 'neutral';
3. Recipe: set of recipes labeled by their region of origin;
4. Ohsumed: collection of medical abstracts;
5. Classic: sets of sentences from academic papers, labeled by publisher name;
6. Reuters: classic news labeled by news topics [4].

Dataset	#docs	Voc	Avg $\|y\|$	
Bbcsport	517	13,243	117	5
Twitter	2,176	6,344	9.9	3
Recipe	3,059	5,708	48.5	15
Ohsumed	3,999	31,789	59.2	10
Classic	4,965	24,277	38.6	4
Reuters	5,485	22,425	37.1	8

Table 2. Datasets' statistics.

		BBCSPORT	TWITTER	RECIPE	OHSUMED	CLASSIC	REUTERS
Unsupervised		LSI	4.30 ± 0.60	31.70 ± 0.70	45.40 ± 0.50	44.20	6.70 ± 0.40	6.30
		WMD	4.60 ± 0.70	28.70 ± 0.60	42.60 ± 0.30	44.50	2.88 ± 0.10	3.50
		Stopword RWMD	4.27 ± 1.19	27.51 ± 1.00	43.98 ± 1.40	44.27	3.25 ± 0.50	5.25
	Cross	All, 5nn	6.00 ± 1.34	29.23 ± 1.09	42.52 ± 1.18	46.73	3.18 ± 0.44	6.26
		All, 5nn, Mean	4.00 ± 1.55	28.58 ± 2.29	42.53 ± 0.67	43.90	3.08 ± 0.62	5.76
		k-means, 5nn	5.91 ± 2.65	28.56 ± 1.20	42.23 ± 1.15	46.50	2.98 ± 0.66	4.71
		k-means, 5nn, Mean	3.82 ± 1.72	28.50 ± 1.51	41.95 ± 1.04	44.05	3.08 ± 0.51	4.57
		ELMo (avg)	6.36 ± 1.24	27.51 ± 1.03	40.66 ± 1.15	68.31	**1.15 ± 0.26**	6.30
Supervised		S-WMD (NCA)	2.10 ± 0.50	27.50 ± 0.50	39.20 ± 0.30	**34.30**	3.20 ± 0.20	3.20
		LMNN	**1.73 ± 0.67**	28.86 ± 2.22	40.88 ± 1.88	39.59	2.76 ± 0.30	4.02
		MCML	2.45 ± 1.27	**27.15 ± 1.36**	**38.93 ± 1.24**	42.38	3.56 ± 0.49	**2.92**

Table 3. Comparison in $k$nn test error(%) to LSI, WMD and S-WMD. Blue shows best results in unsupervised methods and bold indicates best result for a dataset.

Table 2 shows statistics for the training datasets[¶], including the number of inputs (docs) , vocabulary size (Voc), the average number of unique words per document (Avg), and the number of classes $|y|$.

Setup. For comparison purposes, we use the train/test splits provided by [19]. Datasets are pre-processed by removing all words in the SMART stop word list [28], except TWITTER. We make use of the pre-trained version of word embeddings [22]., known to the NLP community as word2vec, offering more than three million words/phrases (from Google News), trained using the skip-gram approach [22]. Words that do not exist in word2vec, are removed. As alternative stopwords, compared to SMART, we use another stopword list [32], consisting of 339 words, which is used in popular libraries like Gensim[‖] and spaCy[**]. In k-means clustering we set a $k = 500$ clusters. In all our proposed methods we use the Relaxed WMD (RWMD) instead of WMD, so that we can scale as well to larger datasets with higher vocabularies.

Results. We evaluate our approaches against WMD [19], LSI [9], and Supervised WMD [14]. We remind that we compare to state-of-the-art distance based methods. The effectiveness of the learned metrics is assessed by the $k$nn classification error.

Table 3 demonstrates results of our proposed "bag-of-tricks" to boost Word Mover's Distance. Our unsupervised approaches achieve superior results in four out of six datasets. Stopword removal with alternative resources can assist the embeddings and reach the Supervised WMD accuracy in the case of the TWITTER dataset. As expected, removing unnecessary stopwords can help WMD significantly, especially in small size documents.

[¶] `https://github.com/mkusner/wmd`

[‖] `https://radimrehurek.com/gensim/`

[**] `https://spacy.io/`

Next, adding neighbors of words that exist in a document, can further enhance the "topical" expression and thus result in better distance computation. We observed that, by incrementing a document with words that are close in the word embedding space, we achieve better accuracy than traditional WMD or LSI approach in most cases. Utilizing prior clustering in word vectors can further boost neighbor words that belong in semantically closer clusters or groups, especially in very small or very large document sizes.

We see that in three datasets, using ELMo as vectors reduced the $k$nn classification error dramatically, with its expressive contextualized power. In the remaining datasets ELMo failed to drop the error, a fact that can be explained since we followed a simple average process over the layers and no fine-tuning was performed. Moreover, we observe that ELMo's worst performance was in OHSUMED, maybe due to big number of classes and specialization of the medical abstracts.

Last, we observe that trying convex metric learning techniques boost the performance of the categorization task in four out of six datasets. As expected, supervised methods yield superior results, with MCML being the best in three datasets. In fact, simple convex loss metric learning resulted in better accuracy, while non-convex NCA can be less stable and accurate due to local minima.

5 Conclusion and Future Work

In this paper, we presented effective and efficient boosting tricks for improving Word Mover's Distance speed and accuracy. We empirically pointed out a number of possible adjustments for the existing WMD, such as playing with different word lists in the stopword removal step, cross document-topic comparison, using deep contextualized word representations, and new metric learning methods. Calibrating those four components (three unsupervised and one supervised), we managed to achieve lower error in the task of text categorization compared to the original WMD and its supervised counterpart.

Measuring similarity between two documents that share words, appearing in different context, can make comparison harder. Thus, the problem of polysemy should also be addressed. In order to address that, topical word embeddings [20] can be applied. Thus, a "topical" WMD, based on topics rather than documents alone, would be a promising direction step. Fine-tuning ELMo for distance computation can be a future direction. Similar contextualized word representations, like BERT [10], can be followed. Moreover, we plan to fully examine non-Linear Metric Learning methods, like Gradient Boosting LMNN or χ^2-LMNN [16] for the supervised version of WMD. Finally, we would like to examine new metrics for measuring distance between text documents, using methodologies from computational geometry.

References

1. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of machine learning research **3**(Feb), 1137–1155 (2003)

2. Bigi, B.: Using kullback-leibler distance for text categorization. In: European Conference on Information Retrieval. pp. 305–319. Springer (2003)
3. Brokos, G.I., Malakasiotis, P., Androutsopoulos, I.: Using centroids of word embeddings and word mover's distance for biomedical document retrieval in question answering. Proceedings of the 15th Workshop on Biomedical Natural Language Processing (2016)
4. Cachopo, A.M.d.J.C.: Improving methods for single-label text categorization. Instituto Superior Técnico, Portugal (2007)
5. Chen, L., Dai, S., Tao, C., Shen, D., Gan, Z., Zhang, H., Zhang, Y., Carin, L.: Adversarial text generation via feature-mover's distance. Advances in Neural Information Processing Systems (2018)
6. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning. pp. 160–167. ACM (2008)
7. Das, R., Zaheer, M., Dyer, C.: Gaussian lda for topic models with word embeddings. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics (2015)
8. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the 24th international conference on Machine learning. pp. 209–216. ACM (2007)
9. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American society for information science **41**(6), 391 (1990)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. NAACL 2019 (2019)
11. Globerson, A., Roweis, S.T.: Metric learning by collapsing classes. In: Advances in neural information processing systems. pp. 451–458 (2006)
12. Goldberger, J., Hinton, G.E., Roweis, S.T., Salakhutdinov, R.R.: Neighbourhood components analysis. In: Advances in neural information processing systems. pp. 513–520 (2005)
13. Harris, Z.S.: Distributional structure. Word **10**(2-3), 146–162 (1954)
14. Huang, G., Guo, C., Kusner, M.J., Sun, Y., Sha, F., Weinberger, K.Q.: Supervised word mover's distance. In: Advances in Neural Information Processing Systems. pp. 4862–4870 (2016)
15. Johnson, R., Zhang, T.: Supervised and semi-supervised text categorization using lstm for region embeddings. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. pp. 526–534. ICML'16, JMLR.org (2016), http://dl.acm.org/citation.cfm?id=3045390.3045447
16. Kedem, D., Tyree, S., Sha, F., Lanckriet, G.R., Weinberger, K.Q.: Non-linear metric learning. In: Advances in Neural Information Processing Systems. pp. 2573–2581 (2012)
17. Kim, S., Fiorini, N., Wilbur, W.J., Lu, Z.: Bridging the gap: Incorporating a semantic similarity measure for effectively mapping pubmed queries to documents. Journal of biomedical informatics **75**, 122–127 (2017)
18. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 1746–1751 (2014), http://aclweb.org/anthology/D/D14/D14-1181.pdf

19. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From word embeddings to document distances. In: ICML (2015)
20. Liu, Y., Liu, Z., Chua, T.S., Sun, M.: Topical word embeddings. In: AAAI. pp. 2418–2424 (2015)
21. Malliaros, F.D., Skianis, K.: Graph-based term weighting for text categorization. In: Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on. pp. 1473–1479. IEEE (2015)
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. ICLR Workshop (2013)
23. Nikolentzos, G., Meladianos, P., Rousseau, F., Stavrakas, Y., Vazirgiannis, M.: Shortest-path graph kernels for document similarity. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 1890–1900 (2017)
24. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)
25. Porter, M.F.: An algorithm for suffix stripping. Program **14**(3), 130–137 (1980)
26. Puurula, A.: Cumulative progress in language models for information retrieval. In: Proceedings of the Australasian Language Technology Association Workshop 2013 (ALTA 2013). pp. 96–100 (2013)
27. Rousseau, F., Vazirgiannis, M.: Graph-of-word and tw-idf: new approach to ad hoc ir. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 59–68. ACM (2013)
28. Salton, G.: The smart retrieval system—experiments in automatic document processing (1971)
29. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM **18**(11), 613–620 (1975)
30. Schofield, A., Magnusson, M., Mimno, D.: Pulling out the stops: Rethinking stopword removal for topic models. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. vol. 2, pp. 432–436 (2017)
31. Skianis, K., Rousseau, F., Vazirgiannis, M.: Regularizing text categorization with clusters of words. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 1827–1837 (2016)
32. Stone, B., Dennis, S., Kwantes, P.J.: Comparing methods for document similarity analysis. TopiCS, DOI **10** (2010)
33. Tao, J., Cuturi, M., Yamamoto, A.: A distance between text documents based on topic models and ground metric learning. The 26th Annual Conference of the Japanese Society for Artificial Intelligence (2012)
34. Van Rijsbergen, C.J.: Information retrieval (1979)
35. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research **10**(Feb), 207–244 (2009)
36. Witt, N., Seifert, C., Granitzer, M.: Explaining topical distances using word embeddings. In: Database and Expert Systems Applications (DEXA), 2016 27th International Workshop on. pp. 212–217. IEEE (2016)
37. Yang, L., Jin, R.: Distance metric learning: A comprehensive survey. Michigan State University **2**(2) (2006)
38. Zhang, M., Liu, Y., Luan, H.B., Sun, M., Izuha, T., Hao, J.: Building earth mover's distance on bilingual word embeddings for machine translation. In: AAAI. pp. 2870–2876 (2016)