An Efficient Approach for Findings Document Similarity Using Optimized Word Mover's Distance

DOI:
10.1007/978-3-031-45170-6_1

Document Version
Final published version

An Efficient Approach for Findings Document Similarity Using Optimized Word Mover's Distance

Atanu Dey[1(✉)], Mamata Jenamani[1], and Arijit De[2]

[1] Indian Institute of Technology Kharagpur, Kharagpur, India
atanu.dey.cse@kgpian.iitkgp.ac.in, mj@iem.iitkgp.ac.in
[2] The University of Manchester, Manchester, UK
arijit.de@manchester.ac.uk

Abstract. We introduce Optimized Word Mover's Distance (OWMD), a similarity function that compares two sentences based on their word embeddings. The method determines the degree of semantic similarity between two sentences considering their interdependent representations. Within a sentence, all the words may not be relevant for determining contextual similarity at the aspect level with another sentence. To account for this fact, we designed OWMD in two ways: first, it decreases system's complexity by selecting words from the sentence pair according to a predefined set of dependency parsing criteria; Second, it applies the *word mover's distance (WMD)* method to previously chosen words. When comparing the dissimilarity of two text sentences, the WMD method is used because it represents the minimal "journey time" required for the embedded words of one sentence to reach the embedded words of another sentence. Finally, adding an exponent function to the inverse of the OWMD dissimilarity score yields the resulting similarity score, called Optimized Word Mover's Similarity (OWMS). Using STSb-Multi-MT dataset, the OWMS measure decreases MSE, RMSE, and MAD error rates by 66.66%, 40.70%, and 37.93% respectively than previous approaches. Again, OWMS reduces MSE, RMSE, and MAD error rates on Semantic Textual Similarity (STS) dataset by 85.71%, 62.32%, and 60.17% respectively. For STSb-Multi-MT and STS datasets, the suggested strategy reduces run-time complexity by 33.54% and 49.43%, respectively, compared to the best of existing approaches.

Keywords: Word embedding · Document distance · Contextual similarity · Document similarity · Word mover's distance · NLP Optimization

1 Introduction

The endeavor of determining how similar in meaning two brief texts are to one another is called "Contextual Similarity" (CS) [1]. Assigning a number between 0 and 1 (or 0 and 5) is a common method of tagging this similarity,

P. Maji et al. (Eds.): PReMI 2023, LNCS 14301, pp. 3–11, 2023.
https://doi.org/10.1007/978-3-031-45170-6_1

with higher scores indicating greater levels of resemblance between the two texts [2]. Numerous research papers have addressed the issue of contextual similarity. Although supervised models perform well in this regard, labeled training data may be costly and fine-tuning hyper-parameter (HP) may be error-prone [4–7]. These issues can be addressed via unsupervised approaches such as ROUGE and BLEU, employed word-matching [8]. These methods also have limitations in terms of computational efficiency, and often miss the information that have been reworded or rearranged from the source text. Unsupervised embedding-based methods have been proposed to tackle this challenge; however, ensemble methods may increase complexity and cost [2]. Instead, word mover's distance (WMD) may be used to evaluate text in a continuous space using pre-trained word embeddings [9–11]. Many applications of WMD have found success, including automatic evaluation of essays, and identification of emotions [1,3]. Such Bag-of-word approaches particularly for lengthy sentences, are computationally costly and may not necessary for aspect-level contextual similarity [1,3]. Figure 1 shows two sample sentence pairs for contextual similarity. The blue-boxed words are recognized utilizing WMD techniques, whereas the green-underlined words are adequate to determine the optimal distance between two contexts.

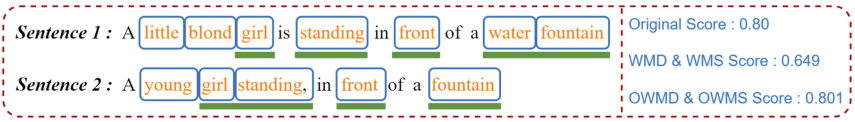

Fig. 1. An example of WMD and OWMD between two sentences

Considering above challenges towards findings aspect-level contextual similarity, the following contributions are as follows for this work: 1) We propose a dependency-parsing-based algorithm to select appropriate words for aspect-level contextual resemblance. It decreases the complexity and improves the accuracy of the system. 2) We have used embedding of aforesaid words as input to the WMD model [1], which is a hyper-parameter (HP) free unsupervised model and less complex than BOW model. 3) Apart from addressing the research challenges, we compare our work with three state-of-the-art methods ROUGE-L, WMS and Re-Eval-WMS on two benchmark datasets such as STSb-Multi-MT and STS.

Rest of the paper is organized as follows. The next section includes a literature review. Section 3 described the proposed method. Section 4 contains the outcomes of the experiment. Section 5 concludes the article and outlines its future applicability.

2 Literature Survey

Table 1 compares modern supervised and unsupervised document similarity methods. Labeled training data is expensive and time-consuming for supervised

Table 1. Comparison on supervised & unsupervised techniques for document similarity

Types	Methodology	Year	Features			Remarks
			HP Tuning	Ensemble	All Data	
Supervised	UWB [4]	2016	✓	✗	✓	Uses deep learning models and natural language processing (NLP) properties (called modified IDF weight).
	BIT [5]	2017	✓	✗	✓	Uses WordNet and British National Corpus to enrich the semantic space.
	ECNU [6]	2017	✓	✓	✓	Builds a global semantic similarity model using ensemble deep learning.
	Learn Short STS [7]	2019	✓	✗	✓	Uses word embeddings and semantic relatedness from other source.
Unsupervised	ROUGE-L [8]	2004	✗	✗	✓	One of the first efforts to employ an expensive Longest Common Word-Matching algorithm b/w two sentences.
	Meerkat Mafia [13]	2014	✓	✗	✓	Trains a Latent Semantic model using three billion English words.
	WMD [1]	2015	✗	✗	✓	Uses bag-of-word (BOW) embeddings to calculate word mover's distance (WMD).
	UESTS [2]	2019	✗	✓	✗	Introduces BabelNet-based synset-focused word aligner.
	WMD & WMS [3]	2019	✗	✗	✓	Uses WMD to compute Word mover's similarity (WMS) score b/w two texts.
	Re-eval WMD [12]	2022	✗	✗	✓	Uses WMD's values in high-dimension spaces, similar to L1-normalized BOW's.
	Proposed	2023	✗	✗	✗	Using dependency-parsing-based algo and WMD to compute Optimized Word mover's similarity (OWMS)

work [4–7]. Unsupervised approach is preferred when labeled training data is not available [1,3]. Several researchers have created unsupervised string-matching algorithms, but if the word sequence changes, they may lose accuracy [2,8]. To address this, a few authors have presented optimization-models for document similarity with accuracy, but they are time-consuming [1,3,12]. High computational cost is of no use for huge volume of datasets. This fact motivates us to simplify a document similarity optimization model without losing accuracy. Word embedding plays a vital role for such optimization models. The 100-dimensional pre-trained Glove vectors [9] beat Word2Vec [11], and BERT [10] on several document similarity datasets. Thus, we choose Glove for this piece of research.

3 Optimized Word Mover's Similarity

Dependency parsing selects words from two target sentences first in the proposed approach. Next, NLP steps remove stop words, symbols, numerical figures, and lemmatization from the selected words. Next, the WMD algorithm determines the optimal distance between the sentences. The detailed procedure is as follows:

3.1 Dependency Parsing Based Word Selection

Without regard to emotional tone, the similarity between two sentences may often be determined by their key contextual words. Therefore, we suggest a dependency parsing-based system for choosing appropriate words. In this regard, we tweak the method used by Qiu et al. (2011), which extracts the sentence's context by selecting just noun phrases [14]. Our deep research reveals that not just noun phrases ('NN', 'NNS') but also certain verb phrases ('VBD', 'VBG', 'VBZ', and 'VB') have contextual meaning at the aspect-level. As we have excluded emotional terms for the sake of similarity, hence no 'adverbs' or 'adjectives' are considered. For example, in Fig. 2, three nouns "girl", "front", "fountain" and one verb "standing" convey the whole meaning of the sentence regardless of other words.

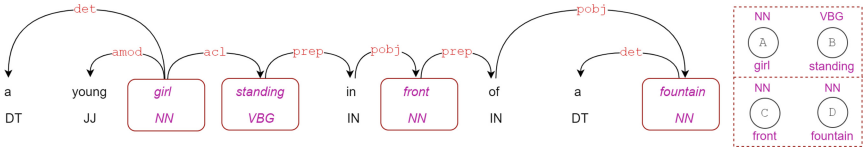

Fig. 2. Example of contextual word selection from a sentence

3.2 Word Mover's Distance (WMD)

WMD calculates text similarity using the discrete *Earth Mover's Distance (EMD)* based on transportation problem [1]. It uses text's bag-of-word (BOW) representations and word embedding similarities. For any two sentences $S1$ and $S2$, WMD is defined as the minimal cost required to transform one into the other. As shown in Eq. 1, the cost amount $F_{i,S1}$ of word i depends on it's relative frequency of the word in a sentence $S1$. Where, $|S1|$ is the total word count of the sentence S1. $F_{j,S2}$ is calculated similarly for sentence $S2$, where index j denotes each word.

$$F_{i,S1} = \frac{count(i)}{|S1|} \tag{1}$$

Now, let w_i represent the embedding of word i, where length of the embedding vector is denoted by d, i.e., $w_i \in \mathbb{R}^d$. The Euclidean distance between embeddings of words i and j is given by $\delta(i, j)$ as shown in Eq. 2.

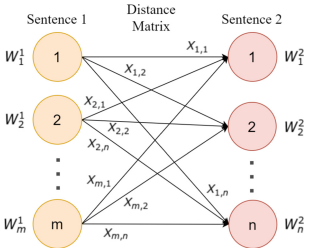

Fig. 3. Sentence to sentence distance measure using selected words and WMD

$$\delta(i,j) \;=\; ||w_i - w_j||^2 \tag{2}$$

Now, WMD can be realized as the solution of linear program as shown in Eq. 3, 4 and 5. Where, m is the number of words in sentence $S1$ and n is the number of words in sentence $S2$ after removing stop words. $\delta(i,j)$ is the Euclidean distance as described in Eq. 2. $X \in \mathbb{R}^{m \times n}$ is a non-negative matrix within which $X_{i,j}$ represents the cost of travelling from word i of sentence $S1$ (denoted by $W_i^1, i = 1..m$) to word j of sentence $S2$ (denoted by $W_j^2, j = 1..n$) as shown in Fig. 3. Specifically, WMD assures that the cost of total outgoing flow cost from each word (i) of $S1$ to all the words (j) in $S2$ is $F_{i,S1}$ in Eq. 4. In addition, Eq. 5 represents the cost of incoming flow to each word (j) of $S2$ from all the words (i) of $S1$ is $F_{j,S2}$.

$$WMD(S1,S2) \;=\; Minimize \sum_{i=1}^{m}\sum_{j=1}^{n} X_{i,j} \,.\, \delta(i,j) \tag{3}$$

subject to:

$$\sum_{j=1}^{n} X_{i,j} = \; F_{i,S1}, \;\forall i \in S1 \tag{4}$$

$$\sum_{i=1}^{m} X_{i,j} = \; F_{j,S2}, \;\forall j \in S2 \tag{5}$$

As shown in Eq. 6, the exponent function transforms the dissimilarity measure (WMD) into a similarity score in the range of {0,1} (called "word mover's similarity" (WMS)), where higher values indicate greater similarity [3].

$$WMS(S1,S2) \;=\; exp\,(-WMD(S1,S2)) \tag{6}$$

We use the above describe process with selective words as explained in Sect. 3.1, which is supported by Hassan et al. [2]. Now, the proposed distance metric is called *optimized word mover's distance (OWMD)* and similarity metric as *optimized word mover's similarity (OWMS)*.

4 Result and Discussion

Following the discussion in Sect. 3, Table 2 shows WMD and OWMD approaches in action. WMD uses BOW technique to extract seven and five words from sentence 1 and 2 respectively, which are highlighted in orange. OWMD uses a dependency parsing (DP) strategy to select five and four words from both sentences, respectively, and underlined them in blue. As discussed in [1] and the pictorial representation of Fig. 3, each of these terms is treated as a node in the linear transportation problem. Multiplying the node counts of both sentences yields the number of decision variables, and adding the node counts yields the number of constraints.

We conduct experiments to compare three state-of-the-art methods with the proposed one using two real-world data sets. Due to a widespread use in various

Table 2. Example of WMD and OWMD methods using two sentences

Methods	Sentence 1	Sentence 2	Variables	Constraints
	a little blonde girl is standing in front of a water fountain	a young girl standing in front of a fountain		
WMD [1]	['little', 'blonde', 'girl', 'standing','front', 'water', 'fountain']	['young','girl', 'standing', 'front', 'fountain']	$7 \times 5 = 35$	$7 + 5 = 12$
OWMD	['girl','stand', 'front', 'water', 'fountain']	['girl','stand', 'front', 'fountain']	$5 \times 4 = 20$	$5 + 4 = 9$

research, the STSb-Multi-MT[1] (2021) and Semantic-Textual-Similarity (STS)[2] (2018) datasets are selected. Both the datasets comprise of 5750 and 13365 sentence pairings with contextual semantic similarity, respectively. Our annotator removes semantically comparable pairs but keeps contextually related ones. Thus, the STSb-Multi-MT and Semantic-Textual Similarity (STS) datasets have 1606 and 3510 sentences, respectively. For instance, "A woman is writing" and "A woman is swimming" should be around 50% similar based on the context of "woman", yet the original dataset solely analyzed semantic activities like "writing" and "swimming", giving just 10% similarity. Our annotators omit such combinations from the experiment because context affects the relevant score. Another pair of sentences, "a man is standing on a roof top playing a violin" and "a man is on a roof dancing", should be $30\% - 40\%$ similar, however the original dataset also gives 36%. Our annotators maintain those couples for the experiment with a context-relevant score. We compare proposed methods with three state-of-the-art methods in terms of three performance metrics: mean-square-error (MSE), root-means-square-error (RMSE), mean absolute deviation (MAD) and three processing times: 1) optimized processing time (OPT) which is responsible for linear programming solution; 2) dependency processing time (DPT) which is responsible for word selection using dependency parsing strategy; 3) total processing time (TPT) is the sum of aforementioned two processing times.

Table 3. Result comparison on STSB-Multi-MT dataset

Methods	**PerformanceMetrics**			**ProcessingTime**		
	MSE	**RMSE**	**MAD**	**OPT**	**DPT**	**TPT**
ROUGE-L (2004) [8]	0.210	0.459	0.363	-	-	-
WMS (2019) [3]	0.012	0.113	0.087	41.043	0.0	41.043
Re-Eval WMS (2022) [12]	0.149	0.387	0.323	41.793	0.0	41.793
OWMS (Ours)	0.004	0.067	0.054	18.423	8.850	27.274
Gain	0.008	0.046	0.033	22.62	-	13.769
Percentage decrease for proposed method	66.66%	40.70%	37.93%	51.11%	-	33.54%

[1] https://huggingface.co/datasets/stsb_multi_mt/viewer/en/train.
[2] https://github.com/anantm95/Semantic-Textual-Similarity/tree/master/data.

Table 4. Result comparison on Semantic Textual Similarity (STS) dataset

Methods	PerformanceMetrics			ProcessingTime		
	MSE	RMSE	MAD	OPT	DPT	TPT
ROUGE-L (2004) [8]	0.192	0.438	0.344	-	-	-
WMS (2019) [3]	0.021	0.146	0.113	106.317	0.0	106.317
Re-Eval WMS (2022) [12]	0.084	0.289	0.229	104.812	0.0	104.812
OWMS (Ours)	0.003	0.055	0.046	34.429	17.829	52.258
Gain	0.018	0.092	0.068	70.383	-	52.554
Percentage decrease for proposed method	85.71%	62.32%	60.17%	66.20%	-	49.43%

Tables 3 and 4 show that our contextual text similarity method outperforms "ROUGE-L", "WMS", and "Re-Eval WMS" on STSb-Multi-MT and Semantic-Textual Similarity (STS) datasets in terms of error evaluation metrics and time complexity. These tables highlight our findings in yellow in the fourth row, while orange represents the best performance among the other three techniques (best among first, second, and third rows). The fifth row shows the proposed technique's gain over the best method. The final row shows the proposed technique's % error and processing time reduction compared to the best-performing prior method. For instance, in Table 3, we minimize MSE, RMSE, and MAD errors by 66.66%, 40.70%, and 37.93%, respectively, compared to the previous approach WMS, which is superior among the three state-of-the-art methods. Our second main claim is that the proposed approach decreases system time complexity compared to prior optimization methods on both the datasets. ROUGE-L is not an optimization algorithm; hence we don't compare it's time complexity to other optimization algorithm. Moreover, in Table 3, we decreased total processing time (TPT) by 13.763 s over the best WMS methodology among two optimization techniques on the STSb-Multi-MT dataset. In Table 4, we reduced TPT by 52.554 s over the best Re-Eval-WMS strategy on the STS dataset. In other words, we reduce TPT by 33.54% and 49.43% on both the dataset respectively. We could have saved 8573.47 seconds (142.89 min or 2.38 h) and 14972.64 seconds (249.54 min or 4.16 h), respectively, if both the datasets contained 1 million identical records.

5 Conclusion

We present OWMD & OWMS, optimized word mover's distance and similarity method for identifying contextual similarity between two sentences at aspect-level irrespective to their semantic matching. This approach has two components: 1) it selects words from sentences using a dependency-parsing based strategy, and 2) it then uses WMD techniques on those words. We conducted experiments using two benchmark datasets namely STSb-Multi-MT and STS. We compare proposed methodology with three contemporary state-of-the-art approaches such

as ROUGE-L, WMS and Re-Eval-WMS. OWMS decreases error rates on STSb-Multi-MT dataset by 66.66%, 40.70%, and 37.93% for MSE, RMSE, and MAD respectively, compared to previous approaches. On STS dataset, OWMS reduces error rates by 85.71%, 62.32%, and 60.17% for MSE, RMSE, and MAD respectively. The proposed solution decreases run-time complexity for STSb-Multi-MT and STS datasets by 33.54 and 49.44%, respectively, compared to previous ones. Thus, it may be used for large datasets containing millions of records and save hours of processing time. The proposed method is especially advantageous in the absence of training data since it is a hyper-parameter-less unsupervised method. In the future, we want to improve the suggested approach and conduct a rigorous mathematical analysis of its robustness.

References

1. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: International Conference on Machine Learning, pp. 957–966. PMLR (2015)
2. Hassan, B., Abdelrahman, S.E., Bahgat, R., Farag, I.: UESTS: an unsupervised ensemble semantic textual similarity method. IEEE Access **7**, 85462–85482 (2019)
3. Clark, E., Celikyilmaz, A., Smith, N.A.: Sentence mover' similarity: automatic evaluation for multi-sentence texts. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2748–2760 (2019)
4. Brychcín, T., Svoboda, L.: UWB at SemEval-2016 task 1: semantic textual similarity using lexical, syntactic, and semantic information. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 588–594 (2016)
5. Wu, H., Huang, H.Y., Jian, P., Guo, Y., Su, C.: BIT at SemEval-2017 task 1: using semantic information space to evaluate semantic textual similarity. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 77–84 (2017)
6. Tian, J., Zhou, Z., Lan, M., Wu, Y.: ECNU at SemEval-2017 task 1: leverage kernel-based traditional NLP features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pp. 191–197 (2017)
7. Nguyen, H.T., Duong, P.H., Cambria, E.: Learning short-text semantic similarity with word embeddings and external knowledge sources. Knowl.-Based Syst. **182**, 104842 (2019)
8. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)
9. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

12. Sato, R., Yamada, M., Kashima, H.: Re-evaluating word mover' distance. In: International Conference on Machine Learning, pp. 19231–19249. PMLR (2022)
13. Kashyap, A.L., et al.: Meerkat mafia: multilingual and cross-level semantic textual similarity systems. In: Proceedings of the 8th International Workshop on Semantic Evaluation, pp. 416–423 (2014)
14. Qiu, G., Liu, B., Jiajun, B., Chen, C.: Opinion word expansion and target extraction through double propagation. Comput. Linguist. **37**(1), 9–27 (2011)