A Appendix A: Proof of Lemma 1 and Theorem 1

A.1 Proof of Lemma 1

Proof. Firstly, we find an ϵ-covering $\mathcal{E}_W$ of size $(\frac{2}{\epsilon})^d$ for the word vector space $\mathcal{V}$. Then define $\mathcal{E}$ as all possible sets of $\boldsymbol{v} \in \mathcal{E}_W$ of size no larger than $L_{\max}$. We have $|\mathcal{E}| \leq (\frac{2}{\epsilon})^{dL_{\max}}$, and for any document $x = (\boldsymbol{v}_j)_{j=1}^L \in \mathcal{X}$, we can find $x_i \in \mathcal{E}$ with also L words $(\boldsymbol{u}_j)_{j=1}^L$ such that $\|\boldsymbol{u}_j - \boldsymbol{v}_j\| \leq \epsilon$. Then by the definition of WMD (1), a solution that assigns each word $\boldsymbol{v}_j$ in x to the word $\boldsymbol{u}_j$ in x_i would have overall cost less than ϵ, and therefore, $\text{WMD}(x, x_i) \leq \epsilon$. □

A.2 Proof of Theorem 1

Proof. Let $s_R(x, y)$ be the random approximation (3). Our goal is to bound the magnitude of $\Delta_R(x, y) = s_R(x, y) - k(x, y)$. Since $E[\Delta_R(x, y)] = 0$ and $|\Delta_R(x, y)| \leq 1$, from Hoefding inequality, we have

$$P\left\{|\Delta_R(x, y)| \geq t\right\} \leq 2\exp(-Rt^2/2)$$

for a given pair of documents (x, y). To get a uniform bound that holds for $\forall(x, y) \in \mathcal{X} \times \mathcal{X}$, we find an ϵ-covering of $\mathcal{X}$ of finite size, given by Lemma 1. Applying union bound over the ϵ-covering $\mathcal{E}$ for x and y, we have

$$\begin{aligned} &P\left\{\max_{x_i \in \mathcal{E}, y_j \in \mathcal{E}} |\Delta_R(x_i, y_j)| > t\right\} \\ &\leq 2|\mathcal{E}|^2 \exp(-Rt^2/2). \end{aligned} \tag{4}$$

Then by the definition of $\mathcal{E}$ we have $|\text{WMD}(x, \omega) - \text{WMD}(x_i, \omega)| \leq \text{WMD}(x, x_i) \leq \epsilon$. Together with the fact that $\exp(-\gamma t)$ is Lipschitz-continuous with parameter γ for $t \geq 0$, we have

$$|\phi_\omega(x) - \phi_\omega(x_i)| \leq \gamma\epsilon$$

and thus

$$|s_R(x, y) - s_R(x_i, y_i)| \leq 3\gamma\epsilon,$$

and

$$|k(x, y) - k(x_i, y_i)| \leq 3\gamma\epsilon$$

for $\gamma\epsilon$ chosen to be ≤ 1. This gives us

$$|\Delta_R(x, y) - \Delta_R(x_i, y_i)| \leq 6\gamma\epsilon \tag{5}$$

Combining (4) and (5), we have

$$\begin{aligned} &P\left\{\max_{x_i \in \mathcal{E}, y_j \in \mathcal{E}} |\Delta_R(x, y)| > t + 6\gamma\epsilon\right\} \\ &\leq 2\left(\frac{2}{\epsilon}\right)^{2dL_{\max}} \exp(-Rt^2/2). \end{aligned} \tag{6}$$

Choosing $\epsilon = t/6\gamma$ yields the result. □

B Appendix B: Additional Experimental Results and Details

B.1 Experimental settings and parameters for WME

Setup. We choose 9 different document corpora where 8 of them are overlapped with datasets in (Kusner et al., 2015; Huang et al., 2016). A complete data summary is in Table 1. These datasets come from various applications, including news categorization, sentiment analysis, product identification, and have various number of classes, varying number of documents, and a wide range of document lengths. Our code is implemented in Matlab and we use the C Mex function for computationally expensive components of Word Mover's Distance [1] (Rubner et al., 2000) and the freely available Word2Vec word embedding [2] which has pre-trained embeddings for 3 millon words/phrases (from Google News) (Mikolov et al., 2013a). All computations were carried out on a DELL dual socket system with Intel Xeon processors 272 at 2.93GHz for a total of 16 cores and 250 GB of memory, running the SUSE Linux operating system. To accelerate the computation of WMD-based methods, we use multithreading with total 12 threads for WME and KNN-WMD in all experiments. For all experiments, we generate random document from uniform distribution with mean centered in Word2Vec embedding space since we observe the best performance with this setting. We perform 10-fold cross-validation to search for best parameters for γ and D_{max} as well as parameter C for LIBLINEAR on training set for each dataset. We simply fix the $D_{min} = 1$, and vary D_{max} in the range of 3 to 21, γ in the range of [1e-2 3e-2 0.10 0.14 0.19 0.28 0.39 0.56 0.79 1.0 1.12 1.58 2.23 3.16 4.46 6.30 8.91 10], and C in the range of [1e-5 1e-4 1e-3 1e-2 1e-1 1 1e1 1e2 3e2 5e2 8e2 1e3 3e3 5e3 8e3 1e4 3e4 5e4 8e4 1e5 3e5 5e5 8e5 1e6 1e7 1e8] respectively in all experiments.

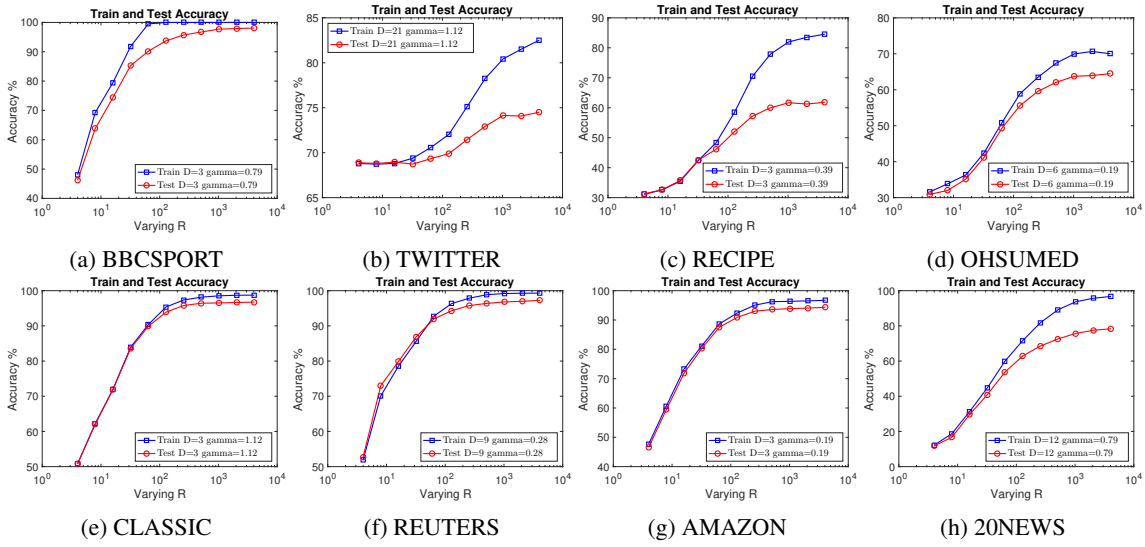
We collect all document corpora from these public websites: BBCSPORT [3], TWITTER [4], RECIPE

[1] We adopt Rubner's C code from http://ai.stanford.edu/~rubner/emd/default.htm.

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

[3] http://mlg.ucd.ie/datasets/bbc.html

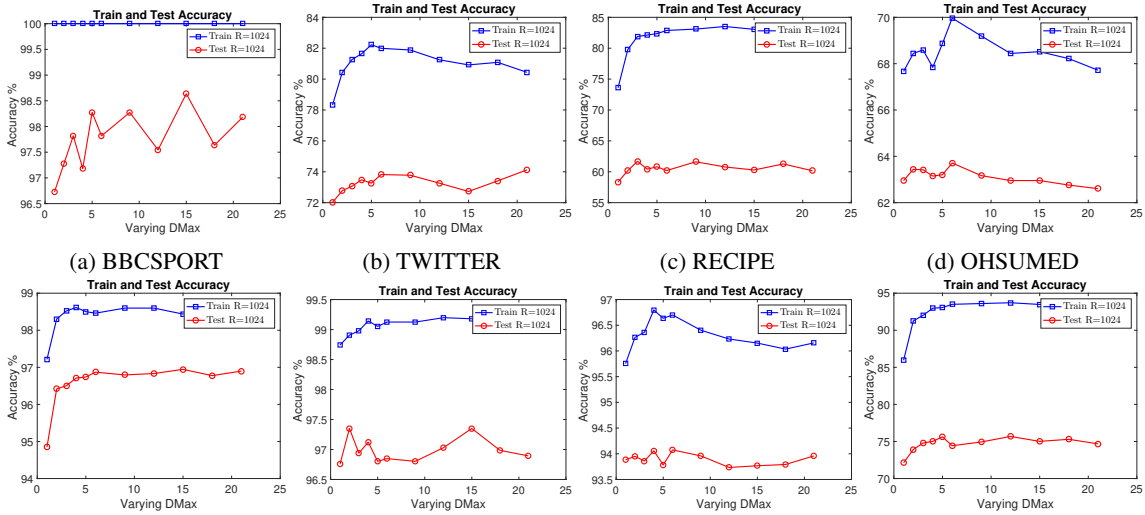
[4] http://www.sananalytics.com/lab/twitter-sentiment/



(a) BBCSPORT (b) TWITTER (c) RECIPE (d) OHSUMED

(e) CLASSIC (f) REUTERS (g) AMAZON (h) 20NEWS

Figure 4: Train (Blue) and test (Red) accuracy when varying R with fixed D.



(a) BBCSPORT (b) TWITTER (c) RECIPE (d) OHSUMED

(e) CLASSIC (f) REUTERS (g) AMAZON (h) 20NEWS

Figure 5: Train (Blue) and test (Red) accuracy when varying D with fixed R.

[5], OHSUMED [6], CLASSIC [7], REUTERS and 20NEWS [8], and AMAZON [9].

B.2 More results about effects of R and D on random documents

Setup and results. To fully study the characteristic of the WME method, we study the effect of the R number of random documents and the D length of random documents on the performance of various datasets in terms of training and testing accuracy. Clearly, the training and testing accuracy can converge rapidly to the exact kernels when varying R from 4 to 4096, which confirms our analysis in Theory 1. When varying D from 1 to 21, we can see that in most of cases $D_{max} = [3,\ 12]$ generally yields a near-peak performance except BBCSPORT.

B.3 More results on Comparisons against distance-based methods

Setup. We preprocess all datasets by removing all words in the SMART stop word list (Buckley et al., 1995). For 20NEWS, we remove the words appear-

[5] https://www.kaggle.com/kaggle/recipe-ingredients-dataset

[6] https://www.mat.unical.it/OlexSuite/Datasets/SampleDataSets-download.htm

[7] http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/

[8] http://www.cs.umb.edu/~smimarog/textmining/datasets/

[9] https://www.cs.jhu.edu/~mdredze/datasets/sentiment/

Table 5: Testing accuracy comparing WME against KNN-based methods

Dataset	BOW	TF-IDF	BM25	LSI	LDA	mSDA	KNN-WMD	WME
BBCSPORT	79.4 ± 1.2	78.5 ± 2.8	83.1 ± 1.5	95.7 ± 0.6	93.6 ± 0.7	91.6 ± 0.8	95.4 ± 0.7	**98.2 ± 0.6**
TWITTER	56.4 ± 0.4	66.8 ± 0.9	57.3 ± 7.8	68.3 ± 0.7	66.2 ± 0.7	67.7 ± 0.7	71.3 ± 0.6	**74.5 ± 0.5**
RECIPE	40.7 ± 1.0	46.4 ± 1.0	46.4 ± 1.9	54.6 ± 0.5	48.7 ± 0.6	52 ± 1.4	57.4 ± 0.3	**61.8 ± 0.8**
OHSUMED	38.9	37.3	33.8	55.8	49.0	50.7	55.5	**64.5**
CLASSIC	64.0 ± 0.5	65.0 ± 1.8	59.4 ± 2.7	93.3 ± 0.4	95.0 ± 0.3	93.1 ± 0.4	**97.2 ± 0.1**	97.1 ± 0.4
REUTERS	86.1	70.9	67.2	93.7	93.1	91.9	96.5	**97.2**
AMAZON	71.5 ± 0.5	58.5 ± 1.2	41.2 ± 2.6	90.7 ± 0.4	88.2 ± 0.6	82.9 ± 0.4	92.6 ± 0.3	**94.3 ± 0.4**
20NEWS	42.2	45.6	44.1	71.1	68.5	60.5	73.2	**78.3**

Table 6: Testing accuracy of WME against Word2Vec and Doc2Vec-based methods.

Dataset	Word2Vec+nbow	Word2Vec+tf-idf	PV-DBOW	PV-DM	Doc2VecC(Train)	Doc2VecC	WME
BBCSPORT	97.3 ± 0.9	96.9 ± 1.1	97.2 ± 0.7	97.9 ± 1.3	89.2 ± 1.4	90.5 ± 1.7	**98.2 ± 0.6**
TWITTER	72.0 ± 1.5	71.9 ± 0.7	67.8 ± 0.4	67.3 ± 0.3	69.8 ± 0.9	71.0 ± 0.4	**74.5 ± 0.5**
OHSUMED	63.0	60.6	55.9	59.8	59.6	63.4	**64.5**
CLASSIC	95.2 ± 0.4	93.9± 0.4	97.0 ± 0.3	96.5 ± 0.7	96.2 ± 0.5	96.6 ± 0.4	**97.1 ± 0.4**
REUTERS	96.9	95.9	96.3	94.9	96.0	96.5	**97.2**
AMAZON	94.0 ± 0.5	92.2 ± 0.4	89.2 ± 0.3	88.6 ± 0.4	89.5 ± 0.4	91.2 ± 0.5	**94.3 ± 0.4**
20NEWS	71.7	70.2	71.0	74.0	72.9	78.2	**78.3**
RECIPE_L	74.9 ± 0.5	73.1 ± 0.6	73.1 ± 0.5	71.1 ± 0.4	75.6 ± 0.4	76.1 ± 0.4	**79.2 ± 0.3**

ing less than 5 times. For LDA, we use the Matlab Topic Modeling Toolbox (Griffiths and Steyvers, 2007) and use sample code that first run 100 burn-in iterations and then run the chain for additional 1000 iterations. For mSDA, we use high-dimensional function mSDAhd where the parameter dd is set as 0.2 times BOW Dimension. For all datasets, a 10-fold cross validation on training set is performed to get the optimal K for KNN classifier, where K is searched in the range of $[1, 21]$.

Baselines. We compare against 7 document representation or distance methods: 1) *bag-of-words* (BOW) (Salton and Buckley, 1988); 2) *term frequency-inverse document frequency* (TF-IDF) (Robertson and Walker, 1994); 3) *Okapi BM25* (Robertson et al., 1995): first TF-IDF variant ranking function used in search engines; 4) *Latent Semantic Indexing* (LSI) (Deerwester et al., 1990): factorize BOW into their leading singular components subspace using SVD (Wu and Stathopoulos, 2015; Wu et al., 2017); 5) *Latent Dirichlet Allocation* (LDA) (Blei et al., 2003): a generative probability method to model mixtures of word "topics" in documents. LDA is trained *transductively* on both train and test; 6) *Marginalized Stacked Denoising Autoencoders* (mSDA) (Chen et al., 2012): a fast method for training denoising autoencoder that achieved state-of-the-art performance on sentiment analysis tasks (Glorot et al., 2011); 7) *WMD*: a state-of-the-art document distance discussed in Section 2.

Results. Table 5 clearly demonstrates the superior performance of our method WME compared to other KNN-based methods in terms of testing accuracy. Indeed, BOW and TF-IDF performs poorly compared to other methods which may be the result of frequent near-orthogonality of their high-dimensional sparse feature representation in KNN classifier. KNN-WMD achieves noticeably better testing accuracy than LSI, LDA and mSDA since WMD takes into account the word alignments and leverages the power of Word2Vec. Remarkably, our proposed method WME achieves much higher accuracy compared to other methods including KNN-WMD on all datasets except one (CLASSIC). The substantially improved accuracy of WME suggests that a truly p.d. kernel implicitly admits expressive feature representation of documents learned from the Word2Vec embedding space in which the alignments between words are considered by using WMD.

B.4 More results on comparisons against Word2Vec and Doc2Vec-based document representations

Setup and results. For *PV-DBOW*, *PV-DM*, and *Doc2VecC*, we set the word and document vector dimension $d = 300$ to match the pre-trained word embeddings we used for WME and other Word2Vec-based methods in order to make a fair comparison. For other parameters, we use recommended parameters in the papers but we search for the best parameter C in LIBLINEAR for these methods. Additionally, we also train *Doc2VecC* with different corruption rate in the range of [0.1 0.3 0.5 0.7 0.9]. Following (Chen, 2017), these

Table 7: Testing accuracy of WME against other document representations on Imdb dataset (50K). Results are collected from (Chen, 2017) and (Arora et al., 2017).

Dataset	RNN_LM	SIF(GloVe)	Word2Vec+AVG	Word2Vec+IDF	PV-DBOW	ST	Doc2VecC	WME
Imdb	86.4	85.0	87.3	88.1	87.9	82.6	88.3	**88.5**

Table 8: Pearson's scores of WME against other unsupervised and supervised methods on 22 textual similarity tasks. Results are collected from (Arora et al., 2017) except our approach.

Approaches	Supervised						Unsupervised					Semi-supervised	
WordEmbeddings	PSL						GloVe					PSL	
Tasks	PP	Dan	RNN	iRNN	LSTM(no)	LSTM(o.g.)	ST	nbow	tf-idf	SIF	WME	SIF	WME
MSRpar	42.6	40.3	18.6	43.4	16.1	9.3	16.8	47.7	**50.3**	35.6	45.3	43.3	49.3
MSRvid	74.5	70.0	66.5	73.4	71.3	71.3	41.7	63.9	77.9	83.8	75.9	**84.1**	76.8
SMT-eur	47.3	43.8	40.9	47.1	41.8	44.3	35.2	46.0	54.7	49.9	**57.7**	44.8	55.6
OnWN	70.6	65.9	63.1	70.1	65.2	56.4	29.7	55.1	64.7	66.2	67.8	**71.8**	69.9
SMT-news	58.4	60.0	51.3	58.1	60.8	51.0	30.8	49.6	45.7	45.6	56.1	53.6	**62.5**
STS'12	58.7	56.0	48.1	58.4	51.0	46.4	30.8	52.5	58.7	56.2	60.6	59.5	**62.8**
headline	72.4	71.2	59.5	72.8	57.4	48.5	34.6	63.8	69.2	69.2	70.5	74.1	**74.2**
OnWN	67.7	64.1	54.6	69.4	68.5	50.4	10.0	49.0	72.9	**82.8**	80.1	82.0	81.9
FNWN	43.9	43.1	30.9	45.3	24.7	38.4	30.4	34.2	36.6	39.4	33.7	**52.4**	32.5
SMT	39.2	38.3	33.8	**39.4**	30.1	28.8	24.3	22.3	29.6	37.9	33.7	38.5	36.7
STS'13	55.8	54.2	44.7	56.7	45.2	41.5	24.8	42.3	52.1	56.6	54.5	**61.8**	56.3
deft forum	48.7	49.0	41.5	49.0	44.2	46.1	12.9	27.1	37.5	41.2	41.2	**51.4**	45.4
deft news	**73.1**	71.7	53.7	72.4	52.8	39.1	23.5	68.0	68.7	69.4	66.7	72.6	69.2
headline	69.7	69.2	57.5	70.2	57.5	50.9	37.8	59.5	63.7	64.7	65.6	70.1	**71.6**
images	78.5	76.9	67.6	78.2	68.5	62.9	51.2	61.0	72.5	82.6	69.2	**84.8**	71.4
OnWN	78.8	75.7	67.7	78.8	76.9	61.7	23.3	58.4	75.2	82.8	81.1	**84.5**	82.3
tweet news	76.4	74.2	58.0	76.9	58.7	48.2	39.9	51.2	65.1	70.1	68.9	**77.5**	68.3
STS'14	70.9	69.5	57.7	70.9	59.8	51.5	31.4	54.2	63.8	68.5	65.5	**73.5**	68.0
answers-forum	68.3	62.6	32.8	67.4	51.9	50.7	36.1	30.5	45.6	63.9	56.4	**70.1**	57.8
answers-student	**78.2**	78.1	64.7	78.1	71.5	55.7	33.0	63.0	63.9	70.4	63.1	75.9	66.2
belief	**76.2**	72.0	51.9	75.9	61.7	52.6	24.6	40.5	49.5	71.8	50.6	75.3	51.6
headline	74.8	73.5	65.3	75.1	64.0	56.6	43.6	61.8	70.9	70.7	70.8	75.9	**76.1**
images	81.4	77.5	71.4	81.1	70.4	64.2	17.7	67.5	72.9	81.5	67.9	**84.1**	69.3
STS'15	75.8	72.7	57.2	75.6	63.9	56.0	31.0	52.7	60.6	71.7	61.8	**76.3**	64.2
SICK'14	71.6	70.7	61.2	71.2	63.9	59.0	49.8	65.9	69.4	72.2	68.0	**72.9**	68.1
Twitter'15	52.9	**53.7**	45.1	52.9	47.6	36.1	24.7	30.3	33.8	48.0	41.6	49.0	47.4

methods are trained transductively on both training and testing set. For *Doc2VecC(Train)*, we train the model only on training set in order to show the effect of the transductive training on the testing accuracy. As shown in Table 6, *Doc2VecC* clearly outperforms *Doc2VecC(Train)*, sometimes having a significant performance boost on some datasets (OHSUMED and 20NEWS).

We further conduct experiments on Imdb dataset using our method. We use only training data to select hyper-parameters. For a more fair comparison, we only report the results of other methods that use all data excluding test. Table 7 shows that WME can achieve slightly better accuracy than other state-of-the-art document representation methods. This collaborates the importance to make full use of both word alignments and high-quality pre-trained word embeddings.

B.5 More results on comparisons for textual similarity tasks

Setup and results. To obtain the hyper-parameters in our method, we use the corresponding training data or the similar tasks from previous years. Note that the tasks with same names but in different years are different ones. As we can see in Table 8, WME can achieve better performance on tasks of STS'12 and perform fairly well on other tasks. Among the unsupervised methods and some supervised methods except *PP*, *Dan*, and *iRNN*, WME is almost always to be one of the best methods.