Determining Gains Acquired from Word Embedding Quantitatively Using Discrete Distribution Clustering

Jianbo Ye*, Yanran Li†, Zhaohui Wu‡, James Z. Wang*, Wenjie Li† and Jia Li*
*The Pennsylvania State University, University Park, Pennsylvania
†The Hong Kong Polytechnic University, Hong Kong ‡Microsoft

Abstract

Word embeddings have become widely-used in document analysis. While a large number of models for mapping words to vector spaces have been developed, it remains undetermined how much net gain can be achieved over traditional approaches based on bag-of-words. In this paper, we propose a new document clustering approach by combining any word embedding with a state-of-the-art algorithm for clustering empirical distributions. By using the Wasserstein distance between distributions, the word-to-word semantic relationship is taken into account in a principled way. The new clustering method is easy to use and consistently outperforms other methods on a variety of data sets. More importantly, the method provides an effective framework for determining when and how much word embeddings contribute to document analysis. Experimental results with multiple embedding models are reported.

1 Introduction

Word embeddings (a.k.a. word vectors) have been broadly adopted for document analysis (Mikolov et al., 2013a,b). The embeddings can be trained from external large-scale corpus and then easily utilized for different data. To a certain degree, the knowledge mined from the corpus, possibly in very intricate ways, is coded in the vector space, the samples of which are easy to describe and ready for mathematical modeling. Despite the appeal, researchers will be interested in knowing how much gain an embedding can bring forth over the performance achievable by existing bag-of-words based approaches. Moreover, how can the gain be quantified? Such a preliminary evaluation will be carried out before building a sophisticated pipeline of analysis.

Almost every document analysis model used in practice is constructed assuming a certain basic representation—bag-of-words or word embeddings—for the sake of computational tractability. For example, after word embedding is done, high-level models in the embedded space, such as entity representations, similarity measures, data manifolds, hierarchical structures, language models, and neural architectures, are designed for various tasks. In order to invent or enhance analysis tools, we want to understand precisely the pros and cons of the high-level models and the underlying representations. Because the model and the representation are tightly coupled in an analytical system, it is not easy to pinpoint where the gain or loss found in practice comes from. Should the gain be credited to the mechanism of the model or to the use of word embeddings? As our experiments demonstrate, introducing certain assumptions will make individual methods effective only if certain constraints are met. We will address this issue under an unsupervised learning framework.

Our proposed clustering paradigm has several advantages. Instead of packing the information of a document into a fixed-length vector for subsequent analysis, we treat a document more thoroughly as a distributional entity. In our approach, the distance between two empirical

Correspondence should be sent to J. Ye (`jxy198@psu.edu`) and J. Li (`jiali@psu.edu`). The work was done when Z. Wu was with Penn State.

nonparametric measures (or discrete distributions) over the word embedding space is defined as the Wasserstein metric (a.k.a. the Earth Mover's Distance or EMD) (Wan, 2007; Kusner et al., 2015). Comparing with a vector representation, an empirical distribution can represent with higher fidelity a cloud of points such as words in a document mapped to a certain space. In the extreme case, the empirical distribution can be set directly as the cloud of points. In contrast, a vector representation reduces data significantly, and its effectiveness relies on the assumption that the discarded information is irrelevant or nonessential to later analysis. This simplification itself can cause degradation in performance, obscuring the inherent power of the word embedding space.

Our approach is intuitive and robust. In addition to a high fidelity representation of the data, the Wasserstein distance takes into account the cross-term relationship between different words in a *principled* fashion. According to the definition, the distance between two documents A and B are the minimum cumulative cost that words from document A need to "travel" to match exactly the set of words for document B. Here, the travel cost of a path between two words is their (squared) Euclidean distance in the word embedding space. Therefore, how much benefit the Wasserstein distance brings also depends on how well the word embedding space captures the semantic difference between words.

While Wasserstein distance is well suited for document analysis, a major obstacle of approaches based on this distance is the computational intensity, especially for the original D2-clustering method (Li and Wang, 2008). The main technical hurdle is to compute efficiently the Wasserstein barycenter, which is itself a discrete distribution, for a given set of discrete distributions. Thanks to the recent advances in the algorithms for solving Wasserstein barycenters (Cuturi and Doucet, 2014; Ye and Li, 2014; Benamou et al., 2015; Ye et al., 2017), one can now perform document clustering by directly treating them as empirical measures over a word embedding space. Although the computational cost is still higher than the usual vector-based clustering methods, we believe that the new clustering approach has reached a level of efficiency to justify its usage given how important it is to obtain high-quality clustering of unstructured text data. For instance, clustering is a crucial step performed ahead of cross-document co-reference resolution (Singh et al., 2011), document summarization, retrospective events detection, and opinion mining (Zhai et al., 2011).

1.1 Contributions

Our work has two main contributions. First, we create a basic tool of document clustering, which is easy to use and scalable. The new method leverages the latest numerical toolbox developed for optimal transport. It achieves state-of-the-art clustering performance across heterogeneous text data—an advantage over other methods in the literature. Second, the method enables us to quantitatively inspect how well a word-embedding model can fit the data and how much gain it can produce over the bag-of-words models.

2 Related Work

In the original D2-clustering framework proposed by Li and Wang (2008), calculating Wasserstein barycenter involves solving a large-scale LP problem at each inner iteration, severely limiting the scalability and robustness of the framework. Such high magnitude of computations had prohibited it from deploying in many real-world applications until recently. To accelerate the computation of Wasserstein barycenter, and ultimately to improve D2-clustering, multiple numerical algorithmic efforts have been made in the recent few years (Cuturi and Doucet, 2014; Ye and Li, 2014; Benamou et al., 2015; Ye et al., 2017).

Although the effectiveness of Wasserstein distance has been well recognized in the computer vision and multimedia literature, the property of Wasserstein barycenter has not been well understood. To our knowledge, there still lacks systematic study of applying Wasserstein barycenter and D2-clustering in document analysis with word embeddings.

A closely related work by Kusner et al. (2015) connects the Wasserstein distance to the word embeddings for comparing documents. Our work differs from theirs in the methodology. We directly pursue a scalable clustering setting rather than construct a nearest neighbor graph based on calculated distances, because the calculation of the Wasserstein distances of all pairs is too expensive to be practical. Kusner et al. (2015) used a lower bound that was less costly to compute in order to prune unnecessary full distance calculation, but the scalability of this modified approach is still limited, an issue to be discussed in Section 4.3. On the other hand, our approach adopts the framework similar to the K-means which is of complexity $O(n)$ per iteration and usually converges within just tens of iterations. The computation of D2-clustering, though in its original form was magnitudes heavier than typical document clustering methods, can now be efficiently carried out with parallelization and proper implementations (Ye et al., 2017).

3 The Method

This section introduces the distance, the D2-clustering technique, the fast computation framework, and how they are used in the proposed document clustering method.

3.1 Wasserstein Distance

Suppose we represent each document d_k consisting m_k unique words by a discrete measure or a discrete distribution, where $k = 1, \dots, N$ with N being the sample size:

$$d_k = \sum\nolimits_{i=1}^{m_k} w_i^{(k)} \delta_{x_i^{(k)}} \ . \qquad (1)$$

Here δ_x denotes the Dirac measure with support x, and $w_i^{(k)} \geq 0$ is the "importance weight" for the i-th word in the k-th document, with $\sum_{i=1}^{m_k} w_i^{(k)} = 1$. And $x_i^{(k)} \in \mathbb{R}^d$, called a support point, is the semantic embedding vector of the i-th word. The 2nd-order Wasserstein distance between two documents d_1 and d_2 (and likewise for any document pairs) is defined by the following LP problem: $W^2(d_1, d_2) :=$

$$\begin{aligned} \min_{\Pi} \quad & \sum\nolimits_{i,j} \pi_{i,j} \|x_i^{(1)} - x_j^{(2)}\|_2^2 \\ \text{s.t.} \quad & \sum\nolimits_{j=1}^{m_2} \pi_{i,j} = w_i, \forall i, \quad \sum\nolimits_{i=1}^{m_1} \pi_{i,j} = w_j, \forall j \\ & \pi_{i,j} \geq 0, \forall i, j \ , \end{aligned} \qquad (2)$$

where $\Pi = \{\pi_{i,j}\}$ is a $m_1 \times m_2$ coupling matrix, and let $\{C_{i,j} := \|x_i^{(1)} - x_j^{(2)}\|_2^2\}$ be transportation costs between words. Wasserstein distance is a true metric (Villani, 2003) for measures, and its best exact algorithm has a complexity of $O(m^3 \log m)$ (Orlin, 1993), if $m_1 = m_2 = m$.

3.2 Discrete Distribution (D2-) Clustering

D2-clustering (Li and Wang, 2008) iterates between the assignment step and centroids updating step in a similar way as the Lloyd's K-means. Suppose we are to find K clusters. The assignment step finds each member distribution its nearest mean from K candidates. The mean of each cluster is again a discrete distribution with m support points, denoted by c_i, $i = 1, \dots, K$. Each mean is iteratively updated to minimize its total within cluster variation. We can write the D2-clustering problem as follows: given sample data $\{d_k\}_{k=1}^N$, support size of means m, and desired number of clusters K, D2-clustering solves

$$\min_{c_1,\dots,c_K} \sum\nolimits_{k=1}^{N} \min_{1\leq i\leq K} W^2(d_k, c_i) \ , \qquad (3)$$

where $c_1, \dots, c_K$ are Wasserstein barycenters. At the core of solving the above formulation is an optimization method that searches the Wasserstein barycenters of varying partitions. Therefore, we concentrate on the following problem. For each cluster, we reorganize the index of member distributions from $1, \dots, n$. The Wasserstein barycenter (Agueh and Carlier, 2011; Cuturi and Doucet, 2014) is by definition the solution of

$$\min_{c} \sum\nolimits_{k=1}^{n} W^2(d_k, c) \ , \qquad (4)$$

where $c = \sum_{i=1}^{m} w_i \delta_{x_i}$. The above Wasserstein barycenter formulation involves two levels of optimization: the outer level finding the minimizer of total variations, and the inner level solving Wasserstein distances. We remark that in D2-clustering, we need to solve multiple Wasserstein barycenters rather than a single one. This constitutes the third level of optimization.

3.3 Modified Bregman ADMM for Computing Wasserstein Barycenter

The recent modified Bregman alternating direction method of multiplier (B-ADMM) algorithm (Ye et al., 2017), motivated by the work by Wang and Banerjee (2014), is a practical choice for computing Wasserstein barycenters. We briefly sketch their algorithmic procedure of this optimization method here for the sake of completeness. To solve for Wasserstein barycenter defined in Eq. (4), the key procedure of the modified Bregman ADMM involves iterative updates of four block of primal variables: the support points of c — $\{x_i\}_{i=1}^m$ (with transportation costs $\{C_{i,j}\}^{(k)}$ for $k = 1, \dots, n$), the importance weights of c — $\{w_i\}_{i=1}^m$, and two sets of split matching variables — $\{\pi_{i,j}^{(k,1)}\}$ and $\{\pi_{i,j}^{(k,2)}\}$, for $k = 1, \dots, n$, as well as Lagrangian variables $\{\lambda_{i,j}^{(k)}\}$ for $k = 1, \dots, n$.

In the end, both $\{\pi_{i,j}^{(k,1)}\}$ and $\{\pi_{i,j}^{(k,2)}\}$ converge to the matching weight in Eq. (2) with respect to $d(c, d_k)$. The iterative algorithm proceeds as follows until c converges or a maximum number of iterations are reached: given constant $\tau \geq 10$, $\rho \propto \frac{\sum_{i,j,k} C_{i,j}^{(k)}}{\sum_{k=1}^{n} m_k m}$ and round-off tolerance $\epsilon = 10^{-10}$, those variables are updated in the following order.

Update $\{x_i\}_{i=1}^{m}$ **and** $\{C_{i,j}^{(k)}\}$ in every τ iterations:

$$x_i := \frac{1}{nw_i} \sum_{k=1}^{n} \sum_{j=1}^{m_k} \pi_{i,j}^{(k,1)} x_j^{(k)}, \forall i, \quad (5)$$

$$C_{i,j}^{(k)} := \|x_i - x_j^{(k)}\|_2^2, \;\; \forall i, j \text{ and } k. \quad (6)$$

Update $\{\pi_{i,j}^{(k,1)}\}$ **and** $\{\pi_{i,j}^{(k,2)}\}$. For each i, j and k,

$$\pi_{i,j}^{(k,2)} := \pi_{i,j}^{(k,2)} \exp\left(\frac{-C_{i,j}^{(k)} - \lambda_{i,j}^{(k)}}{\rho}\right) + \epsilon \,, \quad (7)$$

$$\pi_{i,j}^{(k,1)} := w_j^{(k)} \pi_{i,j}^{(k,2)} \Big/ \left(\sum_{l=1}^{m} \pi_{l,j}^{(k,2)}\right) \,, \quad (8)$$

$$\pi_{i,j}^{(k,1)} := \pi_{i,j}^{(k,1)} \exp\left(\lambda_{i,j}^{(k)} / \rho\right) + \epsilon \,. \quad (9)$$

Update $\{w_i\}_{i=1}^{m}$. For $i = 1, \ldots, m$,

$$w_i := \sum_{k=1}^{n} \frac{\sum_{j=1}^{m_k} \pi_{i,j}^{(k,1)}}{\sum_{i,j} \pi_{i,j}^{(k,1)}} \,, \quad (10)$$

$$w_i := w_i \Big/ \left(\sum_{i=1}^{m} w_i\right) \,. \quad (11)$$

Update $\{\pi_{i,j}^{(k,2)}\}$ **and** $\{\lambda_{i,j}^{(k)}\}$. For each i, j and k,

$$\pi_{i,j}^{(k,2)} := w_i \pi_{i,j}^{(k,1)} \Big/ \left(\sum_{l=1}^{m_k} \pi_{i,l}^{(k,1)}\right) \,, \quad (12)$$

$$\lambda_{i,j}^{(k)} := \lambda_{i,j}^{(k)} + \rho \left(\pi_{i,l}^{(k,1)} - \pi_{i,l}^{(k,2)}\right) \,. \quad (13)$$

Eq. (5)-(13) can all be vectorized as very efficient numerical routines. In a data parallel implementation, only Eq. (5) and Eq. (10) (involving $\sum_{k=1}^{n}$) needs to be synchronized. The software package detailed in (Ye et al., 2017) was used to generate relevant experiments. We make available our codes and pre-processed datasets for reproducing all experiments of our approach.

4 Experimental Results

4.1 Datasets and Evaluation Metrics

We prepare six datasets to conduct a set of experiments. Two short-text datasets are created as follows. (D1) BBCNews abstract: We concatenate the title and the first sentence of news posts from BBCNews dataset[1] to create an abstract version. (D2) Wiki events: Each cluster/class contains a set of news abstracts on the same story such as "2014 Crimean Crisis" crawled from Wikipedia current events following (Wu et al., 2015); this dataset offers more challenges because it has more fine-grained classes and fewer documents (with shorter length) per class than the others. It also shows more realistic nature of applications such as news event clustering.

We also experiment with two long-text datasets and two domain-specific text datasets. (D3) Reuters-21578: We obtain the original Reuters-21578 text dataset and process as follows: remove documents with multiple categories, remove documents with empty body, remove duplicates, and select documents from the largest ten categories. Reuters dataset is a highly unbalanced dataset (the top category has more than 3,000 documents while the 10-th category has fewer than 100). This imbalance induces some extra randomness in comparing the results. (D4) 20Newsgroups "bydate" version: We obtain the raw "bydate" version and process them as follows: remove headers and footers, remove URLs and Email addresses, delete documents with less than ten words. 20Newsgroups have roughly comparable sizes of categories. (D5) BBCSports. (D6) Ohsumed and Ohsumed-full: Documents are medical abstracts from the MeSH categories of the year 1991. Specifically, there are 23 cardiovascular diseases categories.

Evaluating clustering results is known to be nontrivial. We use the following three sets of quantitative metrics to assess the quality of clusters by knowing the ground truth categorical labels of documents: (i) Homogeneity, Completeness, and V-measure (Rosenberg and Hirschberg, 2007); (ii) Adjusted Mutual Information (AMI) (Vinh et al., 2010); and (iii) Adjusted Rand Index (ARI) (Rand, 1971). For sensitivity analysis, we use the homogeneity score (Rosenberg and Hirschberg, 2007) as a projection dimension of other metrics, creating a 2D plot to visualize the metrics of a method along different homogeneity levels. Generally speaking, more clusters leads to higher homogeneity by chance.

[1]BBCNews and BBCSport are downloaded from `http://mlg.ucd.ie/datasets/bbc.html`

4.2 Methods in Comparison

We examine four categories of methods that assume a vector-space model for documents, and compare them to our D2-clustering framework. When needed, we use K-means++ to obtain clusters from dimension reduced vectors. To diminish the randomness brought by K-mean initialization, we ensemble the clustering results of 50 repeated runs (Strehl and Ghosh, 2003), and report the metrics for the ensembled one. The largest possible vocabulary used, excluding word embedding based approaches, is composed of words appearing in at least two documents. On each dataset, we select the same set of Ks, the number of clusters, for all methods. Typically, Ks are chosen around the number of ground truth categories in logarithmic scale.

We prepare two versions of the TF-IDF vectors as the unigram model. The ensembled K-means methods are used to obtain clusters. (1) *TF-IDF* vector (Sparck Jones, 1972). (2) *TF-IDF-N* vector is found by choosing the most frequent N words in a corpus, where $N \in \{500, 1000, 1500, 2000\}$. The difference between the two methods highlights the sensitivity issue brought by the size of chosen vocabulary.

We also compare our approach with the following seven additional baselines. They are (3) *Spectral Clustering (Laplacian)*, (4) *Latent Semantic Indexing (LSI)* (Deerwester et al., 1990), (5) *Locality Preserving Projection (LPP)* (He and Niyogi, 2004; Cai et al., 2005), (6) *Non-negative Matrix Factorization (NMF)* (Lee and Seung, 1999; Xu et al., 2003), (7) *Latent Dirichlet Allocation (LDA)* (Blei et al., 2003; Hoffman et al., 2010), (8) *Average of word vectors (AvgDoc)*, and (9) *Paragraph Vectors (PV)* (Le and Mikolov, 2014). Details on their experimental setups and hyper-parameter search strategies can be found in the Appendix.

4.3 Runtime

We report the runtime for our approach on two largest datasets. The experiments regarding other smaller datasets all terminate within minutes in a single machine, which we omit due to space limitation. Like K-means, the runtime by our approach depends on the number of actual iterations before a termination criterion is met. In the Newsgroups dataset, with $m = 100$ and $K = 45$, the time per iteration is 121 seconds on 48 processors. In Reuters dataset, with $m = 100$ and $K = 20$, the time per iteration is 190 seconds on 24 processors. Each run terminates in around tens of iterations typically, upon which the percentage of label changes is less than 0.1%.

Our approach adopts the Elkan's algorithm (2003) pruning unnecessary computations of Wasserstein distance in assignment steps of K-means. For the Newsgroups data (with $m = 100$ and $K = 45$), our approach terminates in 36 iterations, and totally computes $12,162,717$ ($\approx 3.5\% \times 18612^2$) distance pairs in assignment steps, saving 60% ($\approx 1 - \frac{12,162,717}{36 \times 45 \times 18612}$) distance pairs to calculate in the standard D2-clustering. In comparison, the clustering approaches based on K-nearest neighbor (KNN) graph with the prefetch-and-prune method of (Kusner et al., 2015) needs substantially more pairs to compute Wasserstein distance, meanwhile the speed-ups also suffer from the curse of dimensionality. Their detailed statistics are reported in Table 1. Based on the results, our approach is much more practical as a basic document clustering tool.

Method		EMD counts (%)
Our approach	$d = 400$, $K = 10$	**2.0**
Our approach	$d = 400$, $K = 40$	**3.5**
KNN	$d = 400$, $K = 1$	73.9
KNN	$d = 100$, $K = 1$	53.0
KNN	$d = 50$, $K = 1$	23.4

Table 1: Percentage of total 18612^2 Wasserstein distance pairs needed to compute on the full Newsgroup dataset. The KNN graph based on 1st order Wasserstein distance is computed from the prefetch-and-prune approach according to (Kusner et al., 2015).

4.4 Results

We now summarize our numerical results.

Regular text datasets. The first four datasets in Table 2 cover quite general and broad topics. We consider them to be regular and representative datasets encountered more frequently in applications. We report the clustering performances of the ten methods in Fig. 1, where three different metrics are plotted against the clustering homogeneity. The higher result at the same level of homogeneity is better, and the ability to achieve higher homogeneity is also welcomed. *Clearly, D2-clustering is the only method that shows ro-*

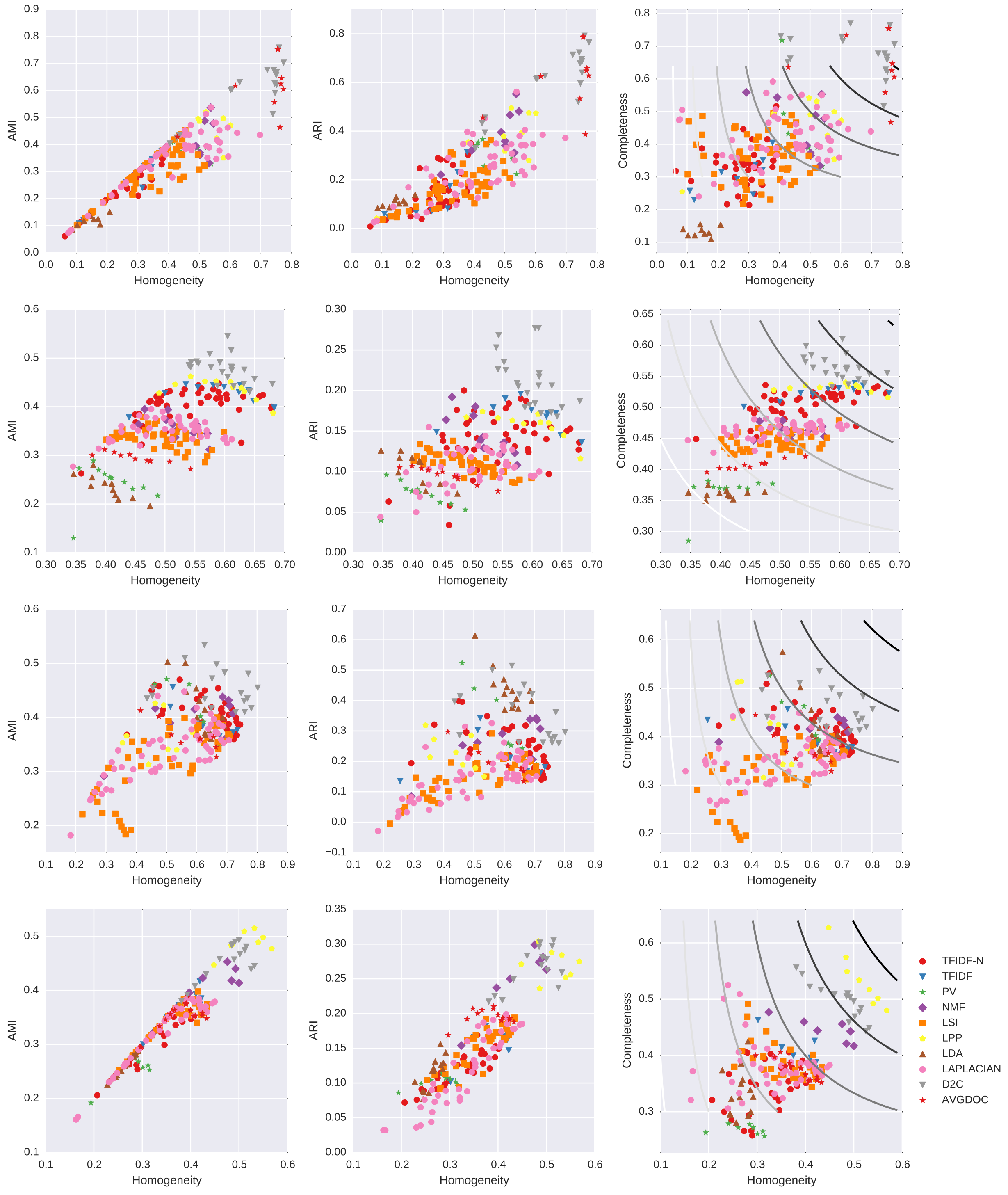

Figure 1: The quantitative cluster metrics used for performance evaluation of "BBC title and abstract", "Wiki events", "Reuters", and "Newsgroups" (row-wise, from top to down). Y-axis corresponds to AMI, ARI, and Completeness, respective (column-wise, from left to right). X-axis corresponds to Homogeneity for sensitivity analysis.

bustly superior performances among all ten methods. Specifically, it ranks first in three datasets, and second in the other one. In comparison, LDA performs competitively on the "Reuters" dataset, but is substantially unsuccessful on others. Meanwhile, LPP performs competitively on the "Wiki events" and "Newsgroups" datasets, but it underperforms on the other two. Laplacian, LSI, and Tfidf-N can achieve comparably performance if their reduced dimensions are fine tuned, which

Dataset	size	class	length	est. #voc.
BBCNews abstr.	2,225	5	26	7,452
Wiki events	1,983	54	22	5,313
Reuters	7,316	10	141	27,792
Newgroups	18,612	20	245	55,970
BBCSports	737	5	345	13,105
Ohsumed	4,340	23	-	-
Ohsumed-full*	34,386	23	184	43,895

Table 2: Description of corpus data that have been used in our experiments. *Ohsumed-full dataset is used for pre-training word embeddings only. Ohsumed is a downsampled evaluation set resulting from removing posts from Ohsumed-full that belong to multiple categories.

unfortunately is unrealistic in practice. NMF is a simple and effective method which always gives stable, though subpar, performance.

Short texts vs. long texts. D2-clustering performs much more impressively on short texts ("BBC abstract" and "Wiki events") than it does on long texts ("Reuters" and "Newsgroups"). This outcome is somewhat expected, because the bag-of-words method suffers from high sparsity for short texts, and word-embedding based methods in theory should have an edge here. As shown in Fig. 1, D2-clustering has indeed outperformed other non-embedding approaches by a large margin on short texts (improved by about 40% and 20% respectively). Nevertheless, we find lifting from word embedding to document clustering is not without a cost. Neither AvgDoc nor PV can perform as competitively as D2-clustering performs on both.

Domain-specific text datasets. We are also interested in how word embedding can help group domain-specific texts into clusters. In particular, does the semantic knowledge "embedded" in words provides enough clues to discriminate fine-grained concepts? We report the best AMI achieved by each method in Table 3. Our preliminary result indicates state-of-the-art word embeddings do not provide enough gain here to exceed the performance of existing methodologies. On the unchallenging one, the "BBCSport" dataset, basic bag-of-words approaches (Tfidf and Tfidf-N) already suffice to discriminate different sport categories; and on the difficult one, the "Ohsumed" dataset, D2-clustering only slightly improves over Tfidf and others, ranking behind LPP. Meanwhile, we feel the overall quality of clustering "Ohsumed" texts is quite far from useful in practice, no matter which method to use. More discussions will be provided next.

4.5 Sensitivity to Word Embeddings.

We validate the robustness of D2-clustering with different word embedding models, and we also show all their results in Fig. 2. As we mentioned, the effectiveness of Wasserstein document clustering depends on how relevant the utilized word embeddings are with the tasks. In those general document clustering tasks, however, word embedding models trained on general corpus perform robustly well with acceptably small variations. This outcome reveals our framework as generally effective and not dependent on a specific word embedding model. In addition, we also conduct experiments with word embeddings with smaller dimensions, at 50 and 100. Their results are not as good as those we have reported (therefore detailed numbers are not included due to space limitation).

Inadequate embeddings may not be disastrous. In addition to our standard running set, we also used D2-clustering with purely random word embeddings, meaning each word vector is independently sampled from spherical Gaussian at 300 dimension, to see how deficient it can be. Experimental results show that random word embeddings degrade the performance of D2-clustering, but it still performs much better than purely random clustering, and is even consistently better than LDA. Its performances across different datasets is highly correlated with the bag-of-words (Tfidf and Tfidf-N). By comparing a pre-trained word embedding model to a randomly generated one, we find that the extra gain is significant ($> 10\%$) in clustering four of the six datasets. Their detailed statistics are in Table 4 and Fig. 3.

5 Discussions

Performance advantage. There has been one immediate observation from these studies, D2-clustering always outperforms two of its degenerated cases, namely Tf-idf and AvgDoc, and three other popular methods: LDA, NMF, and PV, on all tasks. Therefore, for document clustering, users can expect to gain performance improvements by using our approach.

Clustering sensitivity. From the four 2D plots in Fig. 1, we notice that the results of Laplacian,

	regular dataset				domain-specific dataset		
	BBCNews abstract	Wik events	Reuters	Newsgroups	BBCSport	Ohsumed	Avg.
Tfidf-N	0.389	0.448	0.470	0.388	**0.883**	0.210	0.465
Tfidf	0.376	0.446	0.456	0.417	0.799	0.235	0.455
Laplacian	0.538	0.395	0.448	0.385	0.855	0.223	0.474
LSI	0.454	0.379	0.400	0.398	0.840	0.222	0.448
LPP	0.521	0.462	0.426	**0.515**	0.859	**0.284**	0.511
NMF	0.537	0.395	0.438	0.453	0.809	0.226	0.476
LDA	0.151	0.280	0.503	0.288	0.616	0.132	0.328
AvgDoc	**0.753**	0.312	0.413	0.376	0.504	0.172	0.422
PV	0.428	0.289	0.471	0.275	0.553	0.233	0.375
D2C (Our approach)	**0.759**	**0.545**	**0.534**	0.493	0.812	0.260	**0.567**

Table 3: Best AMIs (Vinh et al., 2010) of compared methods on different datasets and their averaging. The best results are marked in bold font for each dataset, the 2nd and 3rd are marked by blue and magenta colors respectively.

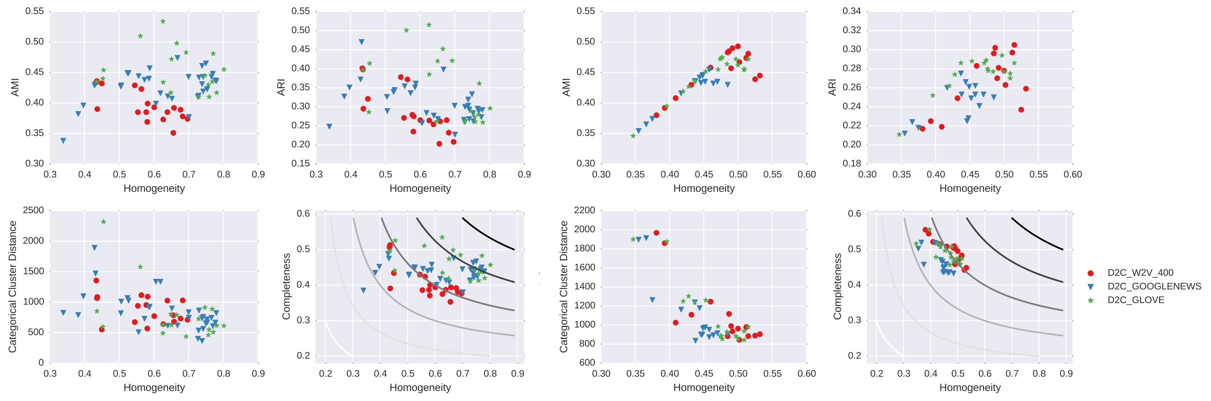

Figure 2: Sensitivity analysis: the clustering performances of D2C under different word embeddings. Left: Reuters, Right: Newsgroups. An extra evaluation index (CCD (Zhou et al., 2005)) is also used.

	ARI	AMI	V-measure
BBCNews abstract	.146	.187	.190
	.792$_{+442\%}$	.759$_{+306\%}$	.762$_{+301\%}$
Wiki events	.194	.369	.463
	.277$_{+43\%}$	.545$_{+48\%}$	.611$_{+32\%}$
Reuters	.498	.524	.588
	.515$_{+3\%}$	.534$_{+2\%}$	.594$_{+1\%}$
Newsgroups	.194	.358	.390
	.305$_{+57\%}$	.493$_{+38\%}$	.499$_{+28\%}$
BBCSport	.755	.740	.760
	.801$_{+6\%}$	.812$_{+10\%}$	.817$_{+8\%}$
Ohsumed	.080	.204	.292
	.116$_{+45\%}$	.260$_{+27\%}$	.349$_{+20\%}$

Table 4: Comparison between *random* word embeddings (upper row) and meaningful *pre-trained* word embeddings (lower row), based on their best ARI, AMI, and V-measures. The improvements by percentiles are also shown in the subscripts.

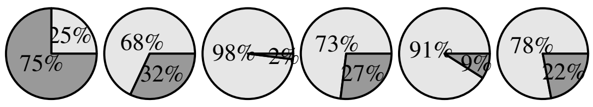

Figure 3: Pie charts of clustering gains in AMI calculated from our framework. Light region is by bag-of-words, and dark region is by pre-trained word embeddings. Six datasets (from left to right): BBCNews abstract, Wiki events, Reuters, Newsgroups, BBCSport, and Ohsumed.

LSI and Tfidf-N are rather sensitive to their extra hyper-parameters. Once the vocabulary set, weight scheme and embeddings of words are fixed, our framework involves only two additional hyper-parameters: the number of intended clusters, K, and the selected support size of centroid distributions, m. We have chosen more than one m in all related experiments ($m = \{64, 100\}$ for long documents, and $m = \{10, 20\}$ for short documents). Our empirical experiments show that the effect of m on different metrics is less sensitive than the change of K. Results at different K are plotted for each method (Fig. 1). The gray dots denote results of multiple runs of D2-clustering. They are always contracted around the top-right region of the whole population, revealing the predictive and robustly supreme performance.

When bag-of-words suffices. Among the results of "BBCSport" dataset, Tfidf-N shows that by restricting the vocabulary set into a smaller one (which may be more relevant to the interest of tasks), it already can achieve highest clustering AMI without any other techniques. Other unsupervised regularization over data is likely unnecessary, or even degrades the performance slightly.

Toward better word embeddings. Our experiments on the Ohsumed dataset have been limited. The result shows that it could be highly desirable to incorporate certain domain knowledge to derive more effective vector embeddings of words and phrases to encode their domain-specific knowledge, such as jargons that have knowledge dependencies and hierarchies in educational data mining, and signal words that capture multi-dimensional aspects of emotions in sentiment analysis.

Finally, we report the best AMIs of all methods on all datasets in Table 3. By looking at each method and the average of best AMIs over six datasets, we find our proposed clustering framework often performs competitively and robustly, which is the only method reaching more than 90% of the best AMI on each dataset. Furthermore, this observation holds for varying lengths of documents and varying difficulty levels of clustering tasks.

6 Conclusions and Future Work

This paper introduces a nonparametric clustering framework for document analysis. Its computational tractability, robustness and supreme performance, as a fundamental tool, are empirically validated. Its ease of use enables data scientists to apply it for the pre-screening purpose of examining word embeddings in a specific task. Finally, the gains acquired from word embeddings are quantitatively measured from a nonparametric unsupervised perspective.

It would also be interesting to investigate several possible extensions to the current clustering work. One direction is to learn a proper ground distance for word embeddings such that the final document clustering performance can be improved with labeled data. The work by (Huang et al., 2016; Cuturi and Avis, 2014) have partly touched this goal with an emphasis on document proximities. A more appealing direction is to develop problem-driven methods to represent a document as a distributional entity, taking into consideration of phrases, sentence structures, and syntactical characteristics. We believe the framework of Wasserstein distance and D2-clustering creates room for further investigation on complex structures and knowledge carried by documents.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. ECCS-1462230, DMS-1521092, and Research Grants Council of Hong Kong under Grant No. PolyU 152094/14E. The primary computational infrastructures used were supported by the Foundation under Grant Nos. ACI-0821527 (CyberStar) and ACI-1053575 (XSEDE).

References

Martial Agueh and Guillaume Carlier. 2011. Barycenters in the Wasserstein space. *SIAM J. Math. Analysis* 43(2):904–924.

Mikhail Belkin and Partha Niyogi. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems (NIPS)*. volume 14, pages 585–591.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. 2015. Iterative Bregman projections for regularized transportation problems. *SIAM J. Sci. Computing (SJSC)* 37(2):A1111–A1138.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *J. Machine Learning Research (JMLR)* 3:993–1022.

Deng Cai, Xiaofei He, and Jiawei Han. 2005. Document clustering using locality preserving indexing. *Trans. Knowledge and Data Engineering (TKDE)* 17(12):1624–1637.

Marco Cuturi and David Avis. 2014. Ground metric learning. *Journal of Machine Learning Research* 15(1):533–564.

Marco Cuturi and Arnaud Doucet. 2014. Fast computation of Wasserstein barycenters. In *Int. Conf. Machine Learning (ICML)*. pages 685–693.

Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *J. American Soc. Information Science* 41(6):391–407.

Charles Elkan. 2003. Using the triangle inequality to accelerate k-means. In *Int. Conf. Machine Learning (ICML)*. volume 3, pages 147–153.

Xiaofei He and Partha Niyogi. 2004. Locality preserving projections. In *Advances in Neural Information Processing Systems (NIPS)*. MIT, volume 16, page 153.

Matthew Hoffman, Francis R Bach, and David M Blei. 2010. Online learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*. pages 856–864.

Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha, and Kilian Q Weinberger. 2016. Supervised word mover's distance. In *Advances in Neural Information Processing Systems (NIPS)*. pages 4862–4870.

Matt J Kusner, Yu Sun, Nicholas N I. Kolkin, and K Q. Weinberger. 2015. From word embeddings to document distances. In *Int. Conf. Machine Learning (ICML)*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Int. Conf. Machine Learning*. pages 1188–1196.

Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755):788–791.

Jia Li and James Z Wang. 2008. Real-time computerized annotation of pictures. *Trans. Pattern Analysis and Machine Intelligence (PAMI)* 30(6):985–1002.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*. pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *HLT-NAACL*. pages 746–751.

James B Orlin. 1993. A faster strongly polynomial minimum cost flow algorithm. *Operations research* 41(2):338–350.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. volume 14, pages 1532–1543.

William M Rand. 1971. Objective criteria for the evaluation of clustering methods. *J. American Statistical Association* 66(336):846–850.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*. volume 7, pages 410–420.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2011. Large-scale cross-document coreference using distributed inference and hierarchical models. In *ACL-HLT*. Association for Computational Linguistics, pages 793–803.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation* 28(1):11–21.

Alexander Strehl and Joydeep Ghosh. 2003. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Machine Learning Research (JMLR)* 3:583–617.

Cédric Villani. 2003. *Topics in optimal transportation*. 58. American Mathematical Soc.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Machine Learning Research (JMLR)* 11:2837–2854.

Xiaojun Wan. 2007. A novel document similarity measure based on earth movers distance. *Information Sciences* 177(18):3718–3730.

Huahua Wang and Arindam Banerjee. 2014. Bregman alternating direction method of multipliers. In *Advances in Neural Information Processing Systems (NIPS)*. pages 2816–2824.

Zhaohui Wu, Chen Liang, and C Lee Giles. 2015. Storybase: Towards building a knowledge base for news events. In *ACL-IJCNLP 2015*. pages 133–138.

Wei Xu, Xin Liu, and Yihong Gong. 2003. Document clustering based on non-negative matrix factorization. In *ACM SIGIR Conf. on Research and Development in Informaion Retrieval*. ACM, pages 267–273.

Jianbo Ye and Jia Li. 2014. Scaling up discrete distribution clustering using admm. In *Int. Conf. Image Processing (ICIP)*. IEEE, pages 5267–5271.

Jianbo Ye, Panruo Wu, James Z. Wang, and Jia Li. 2017. Fast discrete distribution clustering using Wasserstein barycenter with sparse support. *IEEE Trans. on Signal Processing (TSP)* 65(9):2317–2332.

Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. 2011. Clustering product features for opinion mining. In *Int. Conf. on Web Search and Data Mining (WSDM)*. ACM, pages 347–354.

Ding Zhou, Jia Li, and Hongyuan Zha. 2005. A new mallows distance based metric for comparing clusterings. In *Int. Conf. Machine Learning (ICML)*. ACM, pages 1028–1035.

Appendix: Experimental Setups of the Baseline Methods

We select three dimensionality reduction methods. After the dimensionality is reduced, ensembled K-means methods are used to obtain clusters. We remark that most manifold learning approaches, including those we've experimented, are transductive, scaling quadratically w.r.t. the sample size. This makes those approaches difficult to be applied in online or large-scale clustering settings. (3) *Spectral Clustering (Laplacian)*. We apply the Laplacian Eigenmaps (Belkin and Niyogi, 2001) with varying numbers of components to reduce the dimension of Tfidf vectors. Their cosine affinities are constructed based on nearest neighbors. (4) *Latent Semantic Indexing (LSI)* (Deerwester et al., 1990). We compute SVD of the Tfidf matrix, and choose varying numbers of components to form the dimension reduced vectors. (5) *Locality Preserving Projection (LPP)* (He and Niyogi, 2004; Cai et al., 2005). LPP is a popular document indexing method which produces low-dimensional representations. We follow the suggested setup in Cai *et al.* (Cai et al., 2005) and use the cosine affinity matrix.[2] We note that extra hyper-parameters are chosen from a pre-selected set empirically achieving the best performances.

We select two topic models. topic modeling techniques are originally developed to characterize documents with multiple topics, rather than cluster them into disjoint groups. Nevertheless, by assigning each document to its most significant topic, a clustering result can be obtained. We highlight that mixture-of-topics assumption that commonly utilized in topic modeling makes many of their approaches less sensitive to explore the homogeneity of clusters when increasing topics are estimated. (6) *Non-negative Matrix Factorization (NMF)* (Lee and Seung, 1999; Xu et al., 2003). We compute NMF of the Tfidf matrix by choosing K components, where K is the desired number of clusters. Documents (by row) are assigned to their largest column respectively in the factorization matrix to form clusters. (7) *Latent Dirichlet Allocation (LDA)* (Blei et al., 2003; Hoffman et al., 2010). Similar to NMF, we solve a LDA model of the word counting matrix by setting K topics. Because there are many hyper-parameters in a LDA model, our chosen set of hyper-parameters are set to achieve the low perplexity empirically. Adapting LDA to clustering, we naively group documents based on their most likely topics. Empirically, we find it works better than K-means of topic proportion vectors of documents.

Three methods based on word embeddings: we use four pre-trained word embeddings in our experiment to cross validate the effects of different pre-trained word embeddings. Three of them are trained on general large corpora, such as news articles and wikipedia pages. They are 300-dimensional SkipGram (Mikolov et al., 2013a) using negative sampling trained on GoogleNews, 300-dimensional Glove (Pennington et al., 2014) trained on Wikipedia corpus, standard 400-dimensional SkipGram trained on a 2010 Wikipedia dump[3] by our own (with window size of 10 and minimum count of 10). The first two can be downloaded publicly.[4] When compared with other non-embedding methods, best results of the three are reported.

We also use SkipGram to train domain-specific word embeddings using Ohsumed dataset (with window size of 20 and minimum count of 2), which is the fourth model. (8) *Average of word vectors (AvgDoc)* computes the average of the embeddings of distinctive words in a document. In practice, we find it outperforms one using weighted schema like Tfidf or Tf, especially for long texts. (9) *Paragraph Vectors (PV)* (Le and Mikolov, 2014). Two unsupervised methods, *i.e.* PVDM and PVDBOW, are proposed in Le and Mikolov (Le and Mikolov, 2014), in which pre-trained word vectors can be fine-tuned to obtain embeddings for documents. We have experimented with both methods PVDM and PVDBOW, and find PVDM performs significantly worse than PVDBOW on all datasets, thus only the results of PVDBOW are reported.

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/DimensionReduction.html

[3] http://www.psych.ualberta.ca/~westburylab/downloads/westburylab.wikicorp.download.html

[4] https://code.google.com/p/word2vec/ http://nlp.stanford.edu/projects/glove/