WALAILAK JOURNAL
http://wjst.wu.ac.th Information Technology

Short Text Document Clustering using Distributed Word Representation and Document Distance†

Supavit KONGWUDHIKUNAKORN* and Kitsana WAIYAMAI

Department of Computer Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10900, Thailand

(*Corresponding author's e-mail: supavit.kon@ku.th, fengknw@ku.ac.th)

Received: 15 June 2017, Revised: 28 November 2017, Accepted: 19 December 2017

Abstract

This paper presents a method for clustering short text documents, such as instant messages, SMS, or news headlines. Vocabularies in the texts are expanded using external knowledge sources and represented by a Distributed Word Representation. Clustering is done using the K-means algorithm with Word Mover's Distance as the distance metric. Experiments were done to compare the clustering quality of this method, and several leading methods, using large datasets from BBC headlines, SearchSnippets, StackExchange, and Twitter. For all datasets, the proposed algorithm produced document clusters with higher accuracy, precision, F1-score, and Adjusted Rand Index. We also observe that cluster description can be inferred from keywords represented in each cluster.

Keywords: Distributed word representation, document distance, short text documents, short text documents clustering

Introduction

The increasing use of short text documents, including social media posts, SMS, and instant messaging for informative communication has produced a corresponding increase in interest in methods to identify, classify, and make inferences from such content [1]. However, standard methods of text mining and information retrieval perform poorly on such short texts, owing to their conciseness, sparse vocabularies, slangs, frequent grammar errors, incorrect spellings, emoticons, and other unusual characters [2]. Other challenges in extracting latent information from short texts are the rapid evolution of word meanings in social media, words with different meanings depending on context, synonyms, and homonyms of words.

Clustering is a descriptive unsupervised data mining technique that groups data instances into subsets (clusters), such that similar instances are grouped together, while unrelated instances are placed in different subsets [3,4]. The goal is to efficiently partition documents into different subsets based on the similarity of their contexts. Due to the previously mentioned characteristics of short text documents, creating text representations from these documents and clustering them are challenging. A suitable text representation is one of the key factors in improving text clustering efficiency. Research to discover solutions to these issues can be roughly divided into 2 groups of text representation approaches: statistical-based and learning-based.

Among statistical-based text representation methods, term frequency (TF) and term frequency-inverse document frequency (TF-IDF) are statistical weighing schemes that determine the relationship among

†Presented at the 14th International Joint Conference on Computer Science and Software Engineering: July 12th -14th, 2017

words in a set of documents [5,6]. They work by determining the ratio between the frequency of words in specific documents and the inverse proportion of the word over the entire document corpus. TF-IDF measures how specific words are related to particular documents [7]. Although TF-IDF has been widely used in text representation of regular text documents, it does not work very well for short text documents, since the probability of co-occurrences of specific words is low and word sparsity is a major issue [8]. Topic modeling methods, such as latent semantic indexing (LSI) [4,9], probabilistic latent semantic indexing (pLSI) [10], and latent Dirichlet allocation (LDA) [11], address the limitations of TF-IDF by focusing on latent topics in a document corpus with co-occurrences of words. However, these methods generally require at least a few hundred words for accurate determination, making them less suitable for short texts [12,13].

For the learning-based text representation approach, text representations are aggregated with contextual information to increase background knowledge of text documents to solve word sparsity issues in short texts. Numerous research works have addressed the problems of sparsity, such as using convolutional neural networks to predict feature representation and capture local features [8,14,15]. The Dirichlet Multinomial Allocation Model [16] and Gaussian-Bayesian framework are used to perform short text expansion [17], and Gaussian mixture models can capture the notion of latent topics [12]. Quan *et al.* [13], Hong *et al*. [18], Weng *et al*. [19], and Mehrotra *et al.* [20] proposed strategies for adding more context information to documents before applying the learning methods, which has been shown to be necessary and beneficial, such as in Topic Modeling via Self-Aggregation, or in generating long pseudo-documents from short texts before applying the learning methods. These strategies have been demonstrated to cope with short and sparse data issues well. For the deep learning-based text representation approach, the work by Mikolov *et al.* [21] presents a Distributed Representation of Words in a vector space, where each word in a document is represented by a vector of certain size.

In this paper, we introduce a clustering method that uses a Distributed Word Representation constructed using a deep learning text representation approach. This method generates a vector for each vocabulary after being trained on a large corpus of documents, which incorporates some background knowledge of word contexts and relationships based on word co-occurrence. Each word in the text is transformed and represented as a vector. To alleviate word sparsity issues, this text representation preserves semantic relationships between vocabularies in vector space. The K-means algorithm is applied to cluster these vectors into groups of documents with semantically closely-related topics [22]. To measure similarity between documents for topics determination, the Word Mover's Distance (WMD) [23] is applied. WMD is a distance function that measures the dissimilarity between 2 text documents, calculated by aggregating the minimum distances between pairs of words in the 2 documents. After the words are represented in Distributed Representation form, the traditional K-means clustering algorithm is applied to partition texts into different clusters.

The remainder of this paper is organized as follows. First, we describe the background and core concepts of Distributed Representations of Words and Document Distances. Next, our proposed method is presented, followed by a description of the experiments and experimental results. Finally, discussion and conclusions are given in the last section.

Background

To perform text clustering, text documents are generally preprocessed and transformed to text representations. A clustering algorithm is applied to the transformed texts to partition the documents into groups of related concepts. Distributed Word Representation preserves semantic relationships between vocabularies, while Document Distance measures the difference between 2 documents. The text representations and distance functions are described next.

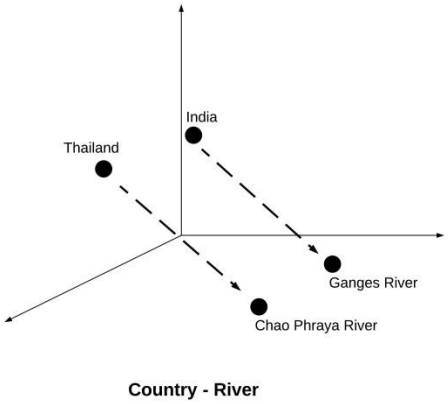

Figure 1 An illustration of vector correlation between a country and the main river in 3-dimensional space. The more correlated vectors are placed closer together, compared to less correlated vectors.

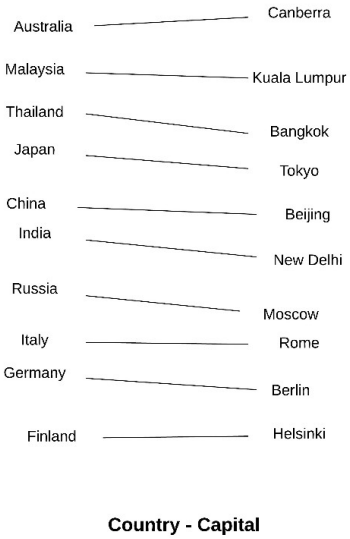

Figure 2 A 2-dimensional projection of vectors of countries and their capital cities.

Distributed word representations

Distributed Word Representations in vector space (also called word embeddings or continuous space representation of words), introduced by Mikolov *et al*. [21,24], provides word vector representation using a neural network model. It is a popular way to capture the semantic similarity between words. The idea is to represent each word in the vocabulary by a vector of certain dimension. The use of word representations was first presented in [25,26]. Subsequent works have developed methods that are more effective in producing text representation by improving training techniques and tools to handle larger vocabulary sizes.

To create the Distributed Representations of Words, the Skip-gram model is used. The Skip-gram model predicts nearby words based on the current word. The model optimizes a network with input,

projection, and output layers in order to maximize the log probability of nearby words in a document. Given a sequence of words $w_1, w_2, \ldots, w_T$, the model is;

$$\frac{1}{T}\sum_{j=1}^{T} \sum_{j\in[-c,c], j\neq 0} \log p\left(w_{t+j} \middle| w_t\right) \quad (1)$$

where c denotes the interval of training context from the center word w_T. A larger value of c enables the model to learn more training words and a more complex syntactic and semantic relationship of words, and thus results in a higher accuracy. $p(w_{t+j}|w_t)$ is the hierarchical softmax function of the word vectors v_{t+j} and v_t. Identifying the Distributed Word Representations is completely unsupervised, and can be trained on any text corpus, or pre-trained in advance. *Word2vec*, a renowned word embedding procedure, is an implementation of the Skip-gram model architecture [21].

The Distributed Word Representations possess special capabilities in automatically identifying and capturing the concepts and semantics of the words to find the correlation between them. Examples of vector representation are shown in **Figure 1**, which represents the associations between a country name and the country's main river. Thailand is closely associated with the Chao Phraya, the country's main river, where India is closely associated with the Ganges. A representation of the association between country name and related capital is shown in **Figure 2**.

Document distance

To group related documents into clusters, a measure (or metric) of document similarity is needed. Such measures include Euclidean distance, Cosine Similarity, and Word Mover's Distance (WMD). WMD, introduced by Kusner *et al.* [23], which is based on the idea of Earth Mover's Distance [27,28], measures how far the words of one document must be "moved" to match another document. To apply WMD, the documents are represented by vectors of some fixed dimension d which contains vocabularies, that is, unique words from the documents. The vocabularies are represented by the matrix $\mathbf{X}$ of size $d \times n$, where d is the number of vector dimension, and n is the number of vocabularies. The i^{th} column, $\mathbf{x}_i \in \mathbb{R}^d$, represents the word embeddings of word i in vector space. The semantic similarity between word i and word j is computed as the distance between them $c(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$, where $c(i,j)$ is the cost of traveling from one word to another. $T_{ij} \in \mathbb{R}^+$ is a flow matrix which indicates how much "mass" of word i in D travels to word j in D'. Thus, the distance between the 2 documents is calculated as the minimum cumulative cost in moving all words from document D to document D', $\sum_{i,j} T_{ij}\, c(i,j)$ [23].

The minimum traveling cost of moving document D to D' is computed by the following formal document transportation equation;

$$\min_{T\in\mathbb{R}^+} \sum_{i,j=1}^{n} T_{i,j}\, c(i,j) \quad (2)$$

Baseline documents representations and distances

The following document representations and document distance methods, in addition to the proposed method, were used in our experiments:

Term Frequency (TF) [29]: A text document representation consisting of the frequency of occurrence of a collection of terms in a text document.

Term Frequency - Inverse Document Frequency (TF-IDF) [29]: A text document representation which uses the term frequencies in the text document divided by the frequency of each term in the entire document corpus.

Euclidean Distance [30]: A standard metric for measuring distance between 2 points, which is also used in text clustering. It is the default distance measure in the K-means clustering algorithm.

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (3)$$

Cosine Similarity [30]: A popular way in finding the correlation between 2 text documents where each document is represented by a vector. The correlation is the cosine of the angle between the 2 vectors. A significant property of Cosine Similarity is the independence of document length.

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} \quad (4)$$

Proposed method

The method begins by creating a pre-trained vector model from a large text corpus. Then, unique vocabularies are generated from short text document datasets. Using the pre-trained vector model, these vocabularies are represented as a vector of fixed dimension. Details of creating the text representation are given in the following subsection. Clustering is performed using the K-means clustering algorithm [31]. The number of desired document clusters depends on the number of different classes in the documents. Documents are assigned to the cluster, with minimum distance between document representation and cluster centroid. Details of Document Distance are given in the subsection on Short Text Similarity Calculation. After each iteration, each cluster centroid is updated based on the similarities of its members. The K-means algorithm is applied iteratively until a standard stopping criterion is satisfied. **Figure 3** shows an overview of the proposed method.

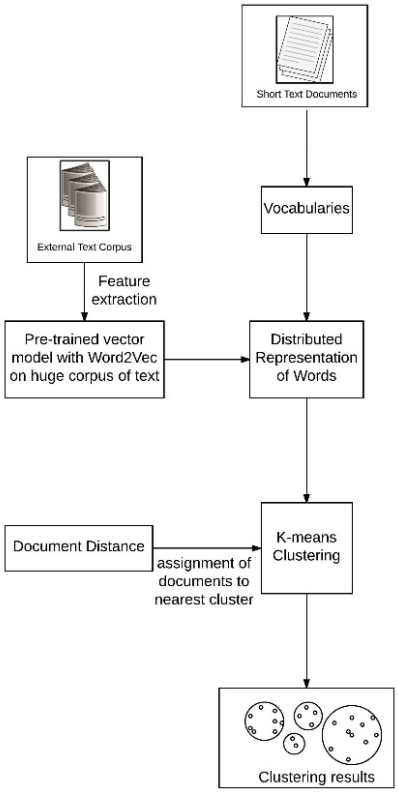

Figure 3 Overview process of the proposed short text clustering method using Document Distance.

Short text representation creation

First, an external large text corpus, in the same domain as the short text documents, is assembled. This corpus is trained using the well-known *word2vec* tool to construct a pre-trained vector model, where each word is represented as a vector of word embeddings. The vectors of similar words are closely placed together, while unrelated words are placed farther apart. The implementation of *word2vec* used is from Python Gensim [32].

The input short text documents are preprocessed by standard text preprocessing techniques, such as removing stopwords and non-informative characters [33,34]. As a result, only unique word vocabularies remain in the short texts. Then, the vector model is applied to these vocabularies to create the word vectors. In this step, short texts are expanded using external knowledge sources. A particular document is represented as a group of aggregated word vectors. An example of creating a short text representation is shown in the first 3 steps of **Figure 4**. The resulting document representations are used for document clustering, as per the next step.

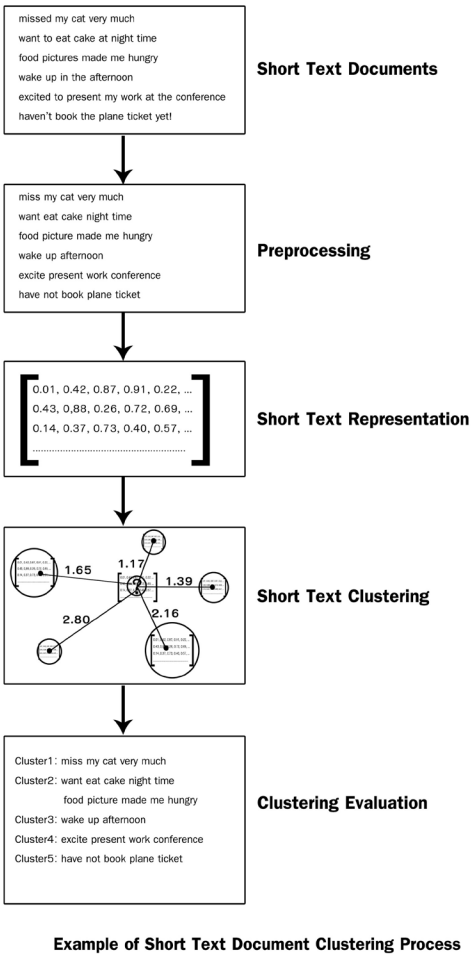

Figure 4 Example of Short Text Document Clustering Process.

Short text similarity calculation

This subsection describes the application of the Word Mover's Distance (WMD) to short text document clustering. WMD measures the dissimilarity of a pair of text documents and the cost of transforming the vocabulary in one document to match another document. In Kusner *et al*. [23], WMD was applied for text classification via the K-nearest neighbor algorithm. In this paper, we apply the same distance for short text clustering.

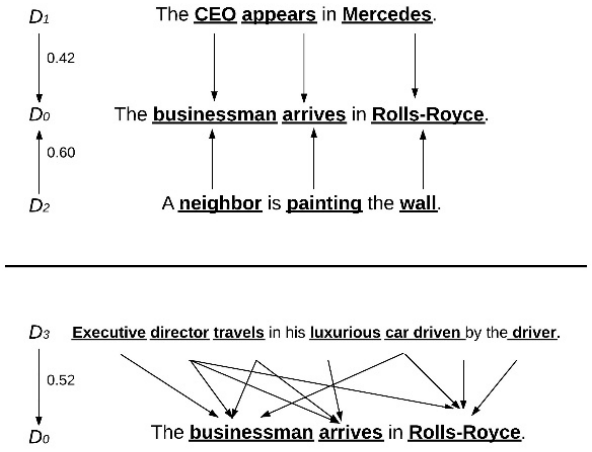

Figure 5 *Top*: The movement in comparing the query sentence D_0 with each of the the 2 sentences D_1 and D_2. These 2 documents have the same bag-of-words distance to D_0. The arrows represent movement between the 2 documents, which are labeled with the cumulative distance of words in each document. *Bottom:* The movement in comparing the query sentence D_0 and the sentence D_3 where the 2 sentences have a different number of words. This causes WMD to compare all pairs of similar words in these sentences. The **underlined bold** words are the vocabulary of the sentence.

To illustrate the application of the Word Mover's Distance in short text similarity calculation, suppose there are 2 sentences, D_1 and D_2, and a referenced query sentence, D_0. To compare the 2 sentences using WMD, stopwords in the sentences must be removed, leaving vocabularies in each sentence. The comparison is made with each pair of sentences, say D_0 and D_1. The vocabularies in D_0 are *businessman, arrives, Rolls-Royce*; while D_1 contains *CEO, appears, Mercedes*. The arrows from each word *i* in D_1 to word *j* in D_0 represent the travel cost. The travel cost of *Mercedes* to *Rolls-Royce* is less than for *wall* to *Rolls-Royce* because the *word2vec* embedding places the vector of *Mercedes* closer to the vector *Rolls-Royce*, the luxury car, than the vector *wall* to the vector *Rolls-Royce*. The travel cost of document *i* to document *j* is the cumulative travel cost of all words in document *i* and document *j*. In this example, the travel cost of D_1 to D_0 (0.42) is smaller than the cost of D_2 to D_0 (0.60). The cost calculation is illustrated in the top part of **Figure 5**. It is perhaps surprising that the algorithm can capture the semantic similarity between documents even though they have no words in common. For comparison, the distances are equal if the bag-of-words/TF-IDF method is used to measure similarity.

Usually, the number of vocabularies in each sentence is not equal, and sequences of vocabularies do not fit the same pattern. As shown in **Figure 5** ***(Bottom)***, sentence D_0 has 3 vocabularies, while sentence D_3 has 7. For document pairs with differing numbers of vocabularies, all pairs of vocabularies in the 2 sentences are used to determine the word pairings that yield the lowest travel cost. To do this, WMD assigns outgoing and incoming weights to each document. These weights, together with the travel costs,

are used in computing the Word Mover's Distance to find the semantic and syntactic similarity of documents. The algorithm is implemented in this work.

To assign a document to the most similar cluster, the Document Distance is measured between the document (represented by a vector) and the centroid vector of each cluster. The document is assigned to the cluster with the smallest document to cluster-centroid distance. An example of short text document similarity calculation is shown in the last 2 steps of **Figure 4**. Circles represent different clusters, while dots in the circles are centroids of the clusters. Each centroid is represented by a vector. The Document Distance is calculated between an un-clustered document, denoted by a circle with a question mark, to the centroid of each cluster. Then, the document is assigned to the cluster with the lowest dissimilarity score (lowest Document Distance). The result of the clustering process is presented in the step of clustering evaluation.

Experimental results

In this section, we describe the application of several methods on 4 public short text document corpora from different domains. We describe the datasets and methodologies that we compared. The performance of different methods is made in terms of clustering quality and clustering outputs.

Evaluation datasets and setup

We conducted experiments using 4 public short text datasets:

BBC News corpus contains headlines of news article from BBC news website collected by Greene and Cunningham [35] for the years 2004 - 2005. The collection consists of 2,225 documents covering 5 areas: *business, entertainment, politics, sport,* and *tech*. For this dataset, we used only news headlines.

SearchSnippets corpus was sampled from the results of web search transactions collected by Phan *et al.* [36], consisting of 12,880 records in 8 different categories. The category labels are *business, computers, culture-arts-entertainment, education-science, engineering, health, politics-society,* and *sports*.

StackExchange corpus contains questions, posts, and comments from the StackExchange communities web forum[1]. We randomly selected short text documents from 8 different categories covering various domains: *3dprinting, AI, aviation, biology, chemistry, physics, security,* and *workplace*. For this dataset, we performed standard text preprocessing techniques [37].

Twitter *tweets* present one of the largest real world user-contributed social media datasets [38]. Texts for this dataset were randomly collected by the authors followed the method described in Boom *et al.* [39]. Python's Tweepy[2] library was used to access the Twitter Streaming API. The data consists of only English Tweets related to 5 different areas: *sleeping, Europe, basketball, technology,* and *weather*. We performed standard text preprocessing techniques as in [37] and basic text clean-up, such as removing symbols, URLs, and user mentions.

The word embeddings are trained using the publicly available *word2vec* tool[3] on a large corpus of documents from a related domain. The parameters are set as in Mikolov *et al.* [21]. Word embeddings of the BBC and SearchSnippets datasets were trained on Wikipedia dumps[4]. The word embeddings of the StackExchange dataset were trained on the whole corpus of StackExchange, which includes questions, posts, and comments. For Twitter, the word embeddings were trained on the entire collected and cleaned corpus.

We performed experiments using 3 text representations: TF, TF-IDF, and Distributed Word Representation. TF and TF-IDF are represented by a vector of word counts with dimension equal to the size of the vocabulary, while the Distributed Representation is generated by the pre-trained vector model based on the short texts dataset. K-means clustering was used to cluster these text representations, where

[1] https://archive.org/details/stackexchange.
[2] https://github.com/tweepy/tweepy.
[3] https://code.google.com/p/word2vec.
[4] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2.

k corresponds to the number of natural classes in the dataset. For clustering, we used 3 distance metrics: Euclidean Distance, Cosine Similarity, and Word Mover's Distance (WMD).

For each dataset, we performed clustering experiments using different combinations of the tree text representation methods and distance metrics, and evaluated the results in terms of cluster quality.

Clustering quality

Clustering quality is measured by accuracy (ACC), precision (PREC), F1-score (F1), and Adjusted Rand Index (ARI) [40]. The performance of the different methods is evaluated on each of the 4 datasets. The 4 statistical measures are computed for each dataset as the average over all clusters of the cluster quality statistics. The results are summarized in **Table 1**. The results show that the Distributed Representation with WMD outperforms all other methods in all clustering quality measurements.

Table 1 Clustering quality in terms of statistical average of accuracy, precision, F1-score, and Adjusted Rand Index.

Method (Dataset: BBC News)	ACC	PREC	F1	ARI
TF + Euclidean	0.255	0.635	0.162	0.003
TF-IDF + Euclidean	0.421	0.474	0.393	0.092
TF + Cosine Similarity	0.425	0.469	0.398	0.105
TF-IDF + Cosine Similarity	0.427	0.473	0.401	0.111
Distributed Representation + Cosine Similarity	0.635	0.865	0.575	0.497
Distributed Representation + WMD	**0.855**	**0.902**	**0.845**	**0.716**
Method (Dataset: SearchSnippets)	ACC	PREC	F1	ARI
TF + Euclidean	0.740	0.910	0.780	0.420
TF-IDF + Euclidean	0.890	0.890	0.890	0.770
TF + Cosine Similarity	0.760	0.780	0.760	0.510
TF-IDF + Cosine Similarity	0.890	0.890	0.890	0.770
Distributed Representation + Cosine Similarity	0.930	0.930	0.800	0.870
Distributed Representation + WMD	**0.990**	**0.990**	**0.990**	**0.970**
Method (Dataset: StackExchange)	ACC	PREC	F1	ARI
TF + Euclidean	0.160	0.640	0.090	0.001
TF-IDF + Euclidean	0.260	0.270	0.250	0.060
TF + Cosine Similarity	0.460	0.500	0.450	0.140
TF-IDF + Cosine Similarity	0.460	0.490	0.470	0.150
Distributed Representation + Cosine Similarity	0.680	0.760	0.660	0.470
Distributed Representation + WMD	**0.770**	**0.850**	**0.760**	**0.550**
Method (Dataset: Twitter)	ACC	PREC	F1	ARI
TF + Euclidean	0.090	0.297	0.033	0.045
TF-IDF + Euclidean	0.183	0.329	0.216	0.160
TF + Cosine Similarity	0.031	0.209	0.031	0.218
TF-IDF + Cosine Similarity	0.126	0.409	0.181	0.064
Distributed Representation + Cosine Similarity	0.867	0.988	0.867	0.959
Distributed Representation + WMD	**0.995**	**0.995**	**0.995**	**0.987**

Table 2 Statistical performance comparison of the proposed method with baseline methods for each clustering quality measurement.

Method: Distributed Representation + WMD vs.	ACC	PREC	F1	ARI
TF + Euclidean	*	*	*	*
TF-IDF + Euclidean	*	*	*	*
TF + Cosine Similarity	*	*	*	*
TF-IDF + Cosine Similarity	*	*	*	*
Distributed Representation + Cosine Similarity	*	*	*	*

**denotes significance*

The Wilcoxon Signed-Rank Test[5] [41] of statistical significance was applied to the statistical metric values in **Table 1** to determine if the differences were statistically significant. The Wilcoxon Signed-Rank Test indicated that the accuracy, precision, F1-score, and ARI scores for clustering using Distributed Representation + WMD were all significantly higher than the other combination methods, with a p-value less than or equal to 0.05. The results are shown in **Table 2**. In this table, * denotes significance.

Table 3 Text datasets characteristics.

Dataset	avg. no. of words	max. no. of words	no. of vocabulary
BBC News	4.4	7	3,712
SearchSnippets	17.9	38	30,646
StackExchange	11.0	160	13,888
Twitter	8.7	21	65,454

Comparing the 3 text representation methods, the Distributed Word Representation method performs better than TF and TF-IDF when applied to short text documents, regardless of the distance metric used for clustering. Furthermore, using WMD as the distance metric with the Distributed Representation outperforms all the other distance metrics. For short text documents, word scantiness is the main concern. Text representations based on statistical weighing schemes do not effectively mitigate this issue. However, the deep learning approach to constructing a text representation, as used by Distributed Representation, seems to substantially alleviate this issue.

Table 3 shows the average number of words per document, maximum number of words per document, and number of vocabulary in each of the 4 text datasets. The BBC News dataset has far fewer words per document (4.4) than other datasets, and also has lower cluster quality than Twitter or SearchSnippets. Clustering quality is not only influenced by document length, but also data cleanness. The low clustering quality for the StackExchange dataset, despite having a greater average number of words than Twitter, illustrates this. Many documents on StackExchange deal with science and mathematics, and contain equation variables in text, which are treated as vocabularies but do not contain semantic value.

Comparing the text similarity metrics, the use of WMD results in significantly better clustering results than Euclidean Distance or Cosine Similarity on all short text representations and datasets. In

[5] Wilcoxon Signed-Rank Test Calculator available online at http://www.socscistatistics.com/tests/signedranks/Default2.aspx

particular, WMD in combination with the Distributed Representation yields the best results, with up to 0.99 for accuracy, precision, F1-score, and Adjusted Rand Index for the Twitter dataset. The nature of the Document Distance function mitigates the text similarity issue.

The results show that Distributed Word Representation improves the performance of document clustering, as applied to short text documents. In summary, representing the short text documents by Distributed Word Representation and measuring similarity by Document Distance outperforms all other methods, as measured by clustering quality statistics.

Clustering outputs

A sample of topics in each cluster from the SearchSnippets dataset is shown in **Figure 6**. Each cluster contains the vocabularies that are contextually and semantically related. Although no cluster is totally pure, it seems reasonable that each cluster's output accurately reflects its members.

In the output of clustering results, each cluster contains the whole short text sentences. Due to limited space, we show only part of the vocabularies in the selected cluster. The 4 clusters shown are for business, computers, education and science, and health contexts. Manual inspection of the sentences in each cluster confirms the accuracy of the results.

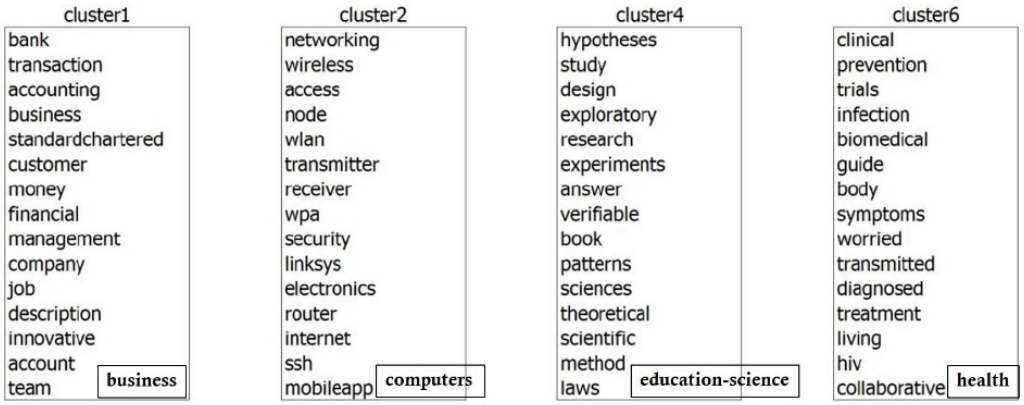

Figure 6 Clustering quality in terms of cluster description.

Conclusion and future works

In this paper, we present an approach to mitigate the effect of text sparsity in clustering short text documents. To achieve this goal, the documents are represented in the form of Distributed Representation of Words expanded by using external knowledge sources. To perform clustering, distance between document vectors is measured using the Word Mover's Distance. Experimental results show that this approach outperforms all other methods in terms of accuracy, precision, F1-scores, and Adjusted Rand Index. In addition, the clustering outputs uncover useful knowledge that captures the keywords and presents the latent topics in these short text documents.

There are still issues that can be improved, such as execution time, which may be improved by efforts to reduce the time complexity. In future work, the proposed approach will also be applied to dynamic and real-time clustering of short text documents.

Acknowledgements

This research was supported by a research scholarship from the Faculty of Engineering, Kasetsart University. Also, we would like to express gratitude to HPCNC laboratory for computational resources used in this study. Special thanks to J. Brucker and T. Kangkachit for their helpful reading and comments on this paper. Thanks to the reviewers for their valuable feedbacks and suggestions.

References

[1] S Liang, E Yilmaz and E Kanoulas. Dynamic clustering of streaming short documents. *In*: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, USA, 2016, p. 995-1004.
[2] A Karandikar. 2010, Clustering short status messages: A topic model based approach, Master's thesis. Faculty of the Graduate School, University of Maryland, Maryland, USA.
[3] L Rokach and O Maimon. *Clustering Methods*. Springer US, Boston, MA, 2005, p. 321-52.
[4] CC Aggarwal and C Zhai. *A Survey of Text Clustering Algorithms*. Springer US, Boston, MA, 2012, p. 77-128.
[5] G Salton and MJ McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, USA, 1986.
[6] KS Jones. A statistical interpretation of term specificity and its application in retrieval. *J. Document.* 1972; **28**, 11-21.
[7] J Ramos. Using tf-idf to determine word relevance in document queries. *In*: Proceedings of the 1st Informational Conference on Machine Learning, Piscataway, USA, 2003.
[8] J Xu, P Wang, G Tian, B Xu, J Zhao, F Wang and H Hao. Short text clustering via convolutional neural networks. *In*: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Colorado, USA, 2015, p. 62-9.
[9] S Deerwester, ST Dumais, GW Furnas, TK Landauer and R Harshman. Indexing by latent semantic analysis. *J. Am. Soc. Inform. Sci.* 1990; **41**, 391-407.
[10] T Hofmann. Probabilistic latent semantic indexing. *In:* Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, USA, 1999, p. 50-7.
[11] DM Blei, AY Ng and MI Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.* 2003; **3**, 993-1022.
[12] VKR Sridhar. Unsupervised topic modeling for short texts using distributed representations of words. *In*: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Denver, Colorado, USA, 2015, p. 192-200.
[13] X Quan, C Kit, Y Ge and SJ Pan. Short and sparse text topic modeling via self-aggregation. *In*: Proceedings of the 24th International Conference on Artificial Intelligence, Buenos Aires, Argentina, 2015, p. 2270-6.
[14] N Kalchbrenner, E Grefenstette and P Blunsom. A convolutional neural network for modelling sentences. *In*: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland, USA, 2014, p. 655-65.
[15] J Xu, B Xu, P Wang, S Zheng, G Tian, J Zhao and B Xu. Self-taught convolutional neural networks for short text clustering. *Neural Netw.* 2017; **88**, 22-31.
[16] Y Yan, R Huang, C Ma, L Xu, Z Ding, R Wang, T Huang and B Liu. Improving document clustering for short texts by long documents via a dirichlet multinomial allocation model. *In*: Proceedings of the 1st International Joint Conference of Web and Big Data, Beijing, China, 2017, p. 626-41.
[17] C MA, Q Zhao, J Pan and Y Yan. Short text classification based on distributional representations of words. *IEICE Trans. Inform. Syst*. 2016; **99**, 2562-5.
[18] L Hong and BD Davison. Empirical study of topic modeling in twitter. *In:* Proceedings of the 1st Workshop on Social Media Analytics, New York, USA, 2010, p. 80-8.
[19] J Weng, EP Lim, J Jiang and Q He. Twitterrank: Finding topic-sensitive influential twitterers. *In*: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, New York, USA, 2010, p. 261-70.
[20] R Mehrotra, S Sanner, W Buntine and L Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. *In*: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, USA, 2013, p. 889-92.

[21] T Mikolov, I Sutskever, K Chen, G Corrado and J Dean. Distributed representations of words and phrases and their compositionality. *In*: Proceedings of the 26th International Conference on Neural Information Processing Systems, Nevada, USA, 2013, p. 3111-9.
[22] J MacQueen. Some methods for classification and analysis of multivariate observations. *In*: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, USA, 1967, p. 281-97.
[23] MJ Kusner, Y Sun, NI Kolkin and KQ Weinberger. From word embeddings to document distances. *In*: Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015, p. 957-66.
[24] T Mikolov, K Chen, G Corrado and J Dean. Efficient estimation of word representations in vector space. *In*: Proceedings of the International Conference on Learning Representations, Scottsdale, Arizona, USA, 2013.
[25] DE Rumelhart, JL McClelland and C PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol. 1, MIT Press, Cambridge, USA, 1986.
[26] JL Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Mach. Learn.* 1991; **7**, 195-225.
[27] Y Rubner, C Tomasi and LJ Guibas. A metric for distributions with applications to image databases. *In*: Proceedings of the Sixth International Conference on Computer Vision, Washington DC, USA, 1998.
[28] Y Rubner, C Tomasi and LJ Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision* 2000; **40**, 99-121.
[29] G Salton and C Buckley. Term-weighting approaches in automatic text retrieval. *Inform. Process. Manag.* 1988; **24**, 513-23.
[30] A Huang. Similarity measures for text document clustering. *In*: Proceedings of the Sixth New Zealand Computer Science Research Student Conference, Christchurch, New Zealand, 2008, p. 49-56.
[31] TS Madhulatha. An overview on clustering methods. *Intell. Data Anal.* 2007; **11**, 583-605.
[32] R Rehurek and P Sojka. Software framework for topic modelling with large corpora. *In:* Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 2010, p. 45-50.
[33] D Sailaja, M Kishore, B Jyothi and N Prasad. An overview of pre-processing text clustering methods. *Int. J. Comput. Sci. Inform. Tech*. 2015; **6**, 3119-24.
[34] AI Kadhim, YN Cheah and NH Ahamed. Text document preprocessing and dimension reduction techniques for text document clustering. *In*: Proceedings of the 4th International Conference on Artificial Intelligence and Applications in Engineering and Technology, Kota Kinabalu, Sabah, Malaysia, 2014, p. 69-73.
[35] D Greene and P Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. *In*: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, Pennsylvania, USA, 2006, p. 377-84.
[36] XH Phan, LM Nguyen and S Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. *In*: Proceedings of the 17th International Conference on World Wide Web, New York, USA, 2008, p. 91-100.
[37] S Vijayarani, MJ Ilamathi and M Nithya. Preprocessing techniques for text mining: An overview. *Int. J. Comput. Sci. Comm. Netw*. 2015; **5**, 7-16.
[38] M Speriosu, N Sudan, S Upadhyay and J Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. *In*: Proceedings of the 1st Workshop on Unsupervised Learning in NLP, Stroudsburg, USA, 2011, p. 53-63.
[39] CD Boom, SV Canneyt, T Demeester and B Dhoedt. Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recogn. Lett.* 2016; **80**, 150-6.
[40] S Wagner and D Wagner. *Comparing Clusterings: An Overview*. Fakultät für Informatik, Universität Karlsruhe, Germany, 2007, p. 1-19.
[41] F Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bull*. 1945; **1**, 80-3.