Bag-of-Concepts: Comprehending Document Representation through Clustering Words in Distributed Representation

Han Kyul Kim
hank@dm.snu.ac.kr

Hyunjoong Kim
hyunjoong@dm.snu.ac.kr

Sungzoon Cho
zoon@snu.ac.kr

December 28, 2015

Abstract

With the advent of doc2vec, distributed representation of documents has quickly established itself as one of the most effective techniques for representing a document in a continuous vector space. It has been successfully applied in solving various text mining problems. Despite its outperforming results, it fails to provide interpretable document vectors as meaning behind each feature remains indescribable. In order to overcome this weakness, this paper proposes the bag-of-concepts method for representing a document vector. This proposed method clusters word vectors generated from word2vec into concept clusters, and uses the occurrences of these concept clusters to represent a document vector. Through this representation, document vectors and their subsequently constructed text mining models can be intuitively interpreted and comprehended.

Keywords: bag-of-concepts, document representation, word2vec clustering

1 Introduction

Most popular document representation methods have often relied on the bag-of-words based approaches [1, 10], through which a document is fundamentally represented by counts of word occurrences within a document. For decades, this approach has been shown to be effective for various text mining tasks [7, 8, 16]. One of its major advantages is that it produces intuitively interpretable document vectors. Each feature of a document vector indicates an occurrence of a specific word within a document. The bag-of-words approach, however, can be problematic when a number of documents being represented are enormous. As a number of documents increase, a number of unique words in the entire document set will also naturally increase. Consequently, not only will the generated document vectors be sparse, but their dimensions will also be huge. As the dimension and the sparsity of the document vectors increase, conventional distance metrics such as Euclidean distance or cosine distance become ineffective in describing the differences between the documents. Consequently, text mining models constructed from

the bag-of-words approach can be unsuccessful in capturing proper difference between high dimensional and sparse document vectors. Although various dimension reduction techniques [4, 6] do exist, these technique lose the innate interpretability of the bag-of-words approach.

To overcome such limitation of the bag-of-words approach, doc2vec model [9], an extension of word2vec [11] method, utilizes contextual information of each word and document to embed document vectors with manageable dimension into a continuous vector space. While context of a word indicates surrounding words for a given word, context of a document is defined as distribution of its composing words. With this contextual information, document vectors with similar context information are located close to each other in the embedded space. Consequently, its performance in document clustering and classification task have previously been reported to be better than those of the bag-of-words based models [3]. However, each feature of document vectors generated from doc2vec is difficult to interpret as its value indicates the weight of the neural network used to train doc2vec.

Despite the outstanding representational performance of doc2vec, having a good representation itself is not the ultimate goal of text mining. In order to apply such method in real text mining tasks, document vectors, similar to those produced by the bag-of-words method, need to be interpretable. Interpretable document vectors can provide deeper understanding of a data set and the operating logic behind subsequently constructed text mining models. However, document vectors generated from doc2vec model fails to provide any intuitive interpretability.

In order to compensate for this limitation of doc2vec, this paper suggests the bag-of-concepts approach for representing a document vector. Through clustering distributed representation of words generated from word2vec, this proposed method can maintain representational superiority of the distributed representation, while simultaneously providing vector interpretability and model explainability. With vector interpretability, we can intuitively understand the features and the components of generated document vectors. With model explainability, we can easily comprehend the operating logic behind a text mining model trained with the document vectors generated from the proposed method. This paper has performed document clustering and classification on Reuter dataset to provide both quantitative and qualitative analysis of the proposed method. The results of these tasks are promising, indicating that the proposed method is indeed a realistic alternative method for document representation.

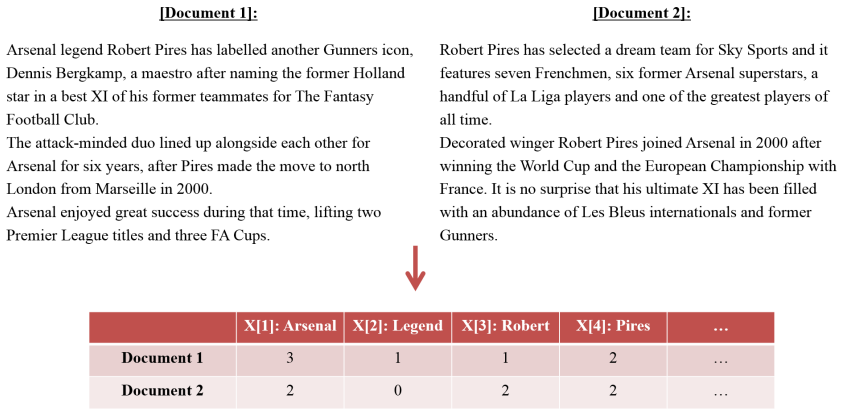
The rest of this paper is structured as follows. In Section 2, we discuss various techniques for document representation in detail. In Section 3 and 4, we propose our word2vec clustering method and describe the dataset used throughout this paper. In Section 5, we provide experiment result of our proposed method to substantiate its vector interpretability and model explainability. We conclude in Section 6 with some discussion and directions for future work.

2 Background

We will discuss three document representation techniques: bag-of-words, word2vec based approach and doc2vec. We provide general idea and motivation behind these methods, and discuss their advantages and disadvantages.

2.1 Bag-of-Words

The bag-of-words approach is established upon an assumption that frequencies of words in a document can indicate the relevance between the documents. Consequently, the features of the document vectors generated from the bag-of-words approach represent the occurrences of each word in a document as shown in Figure 1.



	X[1]: Arsenal	X[2]: Legend	X[3]: Robert	X[4]: Pires	…
Document 1	3	1	1	2	…
Document 2	2	0	2	2	…

Figure 1: Document vectors generated via bag-of-words approach

Due to such explicit features, document vectors generated from the bag-of-words approach can be easily interpreted. If two documents from Figure 1 are calculated to be similar documents, the reason behind their similarity can be explained by directly observing and comparing the features of each document vector. These two document vectors, for example, can be perceived as similar due to the fact that they share similar number of word occurrences for the words "Arsenal", "Legend", "Robert", and "Pires." Consequently, we can reasonably accept the fact that these two document vectors are similar as they both discuss about a same player from a specific football team.

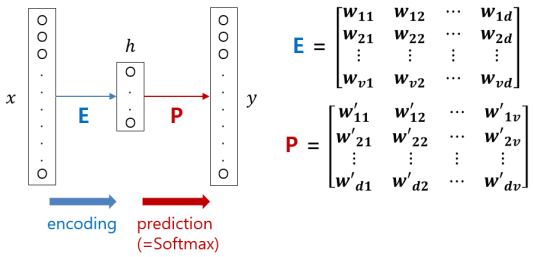
Due to this intuitive interpretability of the generated document vectors, the bag-of-words approach has established itself as one of most influential document representation methods. However, the number of features in these vectors increases significantly as the number of document increases in order to incorporate all of word the occurrences within a set of documents. Consequently, the dimension of the bag-of-words approach can become extremely large and sparse. As the dimension and the sparsity of the document vectors increase, the curse of the dimensionality occurs and conventional distance metrics such as Euclidean distance or cosine

distance become meaningless. Due to such limitation, text mining model constructed from the bag-of-words based document vectors can fail to capture the true differences and similarities between the documents. Although various dimension reduction techniques [4, 6] do exist, these techniques unfortunately lose the innate interpretability of the bag-of-words approach.

2.2 Word2Vec

Although word2vec is a word representation method, it can be expanded into representing documents without much significant modification. Thus, we will first discuss word2vec prior to discussing word2vec based document representation and doc2vec.

Word2vec is based on the assumption of the distributed hypothesis [5], which states that words that occur in similar contexts tend to have similar meanings [14]. Based on this assumption, word2vec uses a simple neural network to embed words into continuous vector space. Through training the weights of the network, word2vec model predicts neighboring words within certain window size for an input word.



$x \in R^{V \times 1}$	One hot representation of a word. Size of input layer is equal to the number of terms
$h \in R^{d \times 1}$	Encoded word vector
$y \in R^{V \times 1}$	Context prediction prob. Size of output layer is also equal to the number of terms
$E \in R^{V \times d}$	Embedding Matrix
w_{ij}	Coordinate of embedding vector. Weight between the ith input node and jth hidden node
$P \in R^{d \times V}$	Context Matrix
w'_{ij}	Coordinate of context vector. Weight between the ith hidden node and jth output node

Figure 2: Word2vec basic architecture

As shown in Figure 2, the size of input layer x is V, equivalent to the total number of unique words in a document set. Each node of the input layer represents an individual word through one-hot encoding. Through encoding matrix E, which essentially is an aggregate of each input node's weight to each of hidden nodes, each word is embedded and represented by the hidden layer h. Consequently, the number of hidden nodes d denotes the dimension of the word vectors and the embedding space. The encoded word vector h is subsequently processed through a respective context vector of context matrix P, which is again an aggregate of each hidden node's weight to each of the node in the output layer. Through P, input word's surrounding context words are predicted with soft-max function that aims at maximizing the cross product between the input word's embedded vector and context vector. Then, this predicted probability of each

word is represented by the value of each node in the output layer y. Identical to the input layer, the size of the output layer y is once again V. By checking whether the predicted context words actually occurred around the input word, accuracy of the prediction is evaluated. Through back-propagation, the values (weights) of the embedding vectors and the context vectors are updated. This general description for training word2vec is depicted in Figure 3.

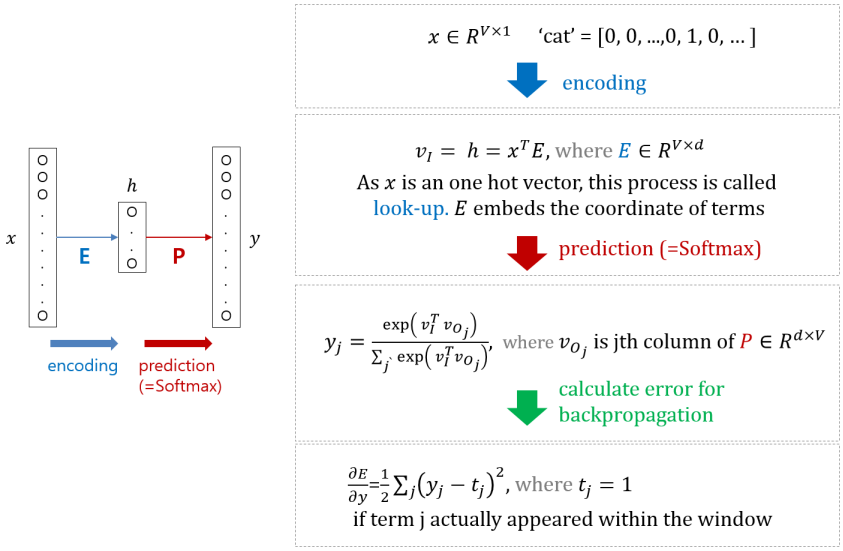


Figure 3: Word2vec training

One of the biggest contributions of word2vec is that the words that occur in similar context - consequently with similar meaning according to the distributed hypothesis - are located close to each other in the embedded space, preserving the semantic similarities between words. As words are represented in a continuous embedded space, various conventional machine learning and data mining techniques can be applied in this space to resolve various text mining tasks [2, 12, 13]. Figure 4 shows an example of such embedded space visualized by t-sne [15]. In this figure, we have embedded words that represent the names of baseball players, the names of football players and the names of countries. While the words with similar meaning are located closer to each other, the words with different meanings are located distant from each other.

Compared to the bag-of-words approach, in which dimension and sparsity of a document vector can increase significantly, word2vec model can be utilized to construct dense document vectors with reasonable number of dimensions. One of the most simple approach for representing a document using word2vec is averaging the word vectors of the words that occurred in a document [17]. Despite its simplicity, its performance in document classification task is shown to be quite promising.

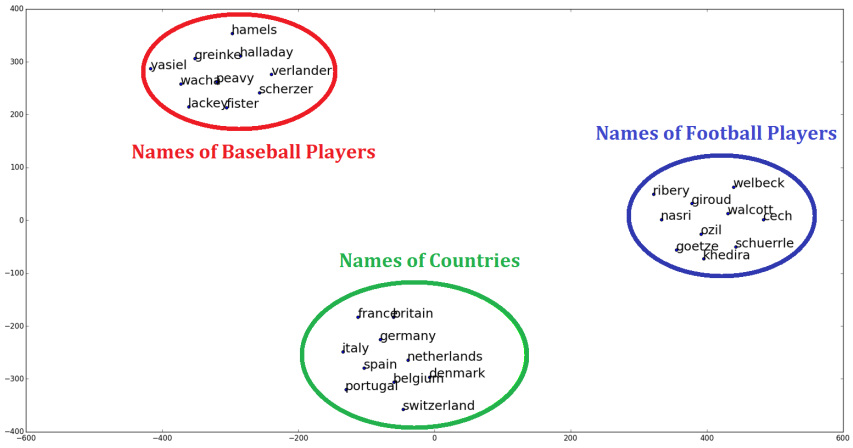
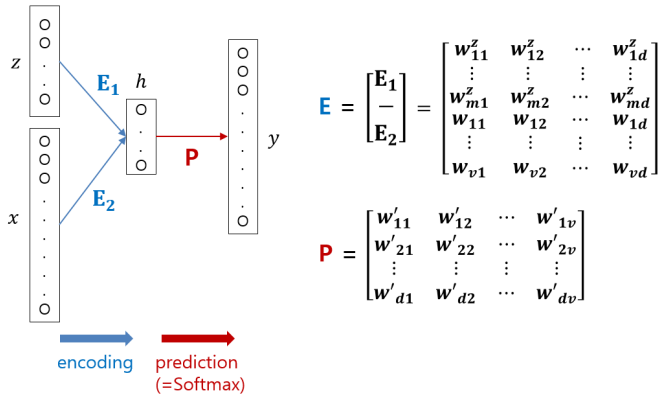


Figure 4: Embedded space using t-sne

2.3 Doc2Vec

Instead of averaging the embedded word2vec vectors to represent a document vector, doc2vec directly embeds documents along with their words as shown in Figure 5.



$z \in R^{m \times 1}$ One hot representation of a document. Size of input layer is equal to the number of documents

w_{ij}^{z} Coordinate of document embedding vector. Weight between the ith input node of z and jth hidden node

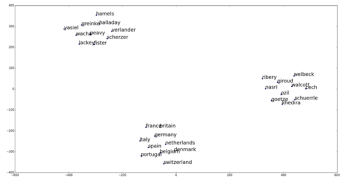
Figure 5: Doc2Vec Architecture

The architecture and the training of the neural network in doc2vec are essentially identical to those of word2vec. The only difference lies in the fact that documents are also incorporated into the network. Similar to words in word2vec model, documents are represented by one hot encoding and embedded into a continuous space through an embedding matrix. As shown in Figure 5, E_1 represents an embedding matrix for documents, while E_2 indicates an embedding matrix for words. Their coordinates, the values of the weight towards the hidden nodes, are similarly updated by back-propagation.

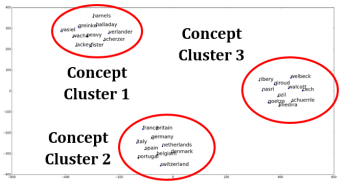
The representation power of doc2vec has been shown to be effective in document clustering and classification tasks [3]. Although the dimensions of document vectors generated from

doc2vec are generally smaller than that of the bag-of-words approach, these features sufficiently incorporate contextual information of words and documents, consequently outperforming the bag-of-words based models. Despite its effective representation power, doc2vec model fails to provide intuitive interpretation behind its generated document vectors. Since each document vector is trained through a neural network, each value of the vector represents only the strength of the connection between an input node and a specific hidden node. Consequently, it is hard to comprehend what exactly each feature of a document vector represents in terms of the contents of a document. Therefore, if a text mining model such as document classifier is trained from these document vectors generated from doc2vec, it fails to provide intuitive explanation for the operating logic behind the model. Having a good representation of a document itself is not be the ultimate goal of text mining. In order for these representation methods to have meaningful impact and implication in real business settings, it is essential that document representation should be able to provide clear understanding and intuition behind the representation and its subsequent text mining model constructed from such representation.

3 Proposed Method



1. Train word2vec to generate word vectors from a collection of documents



2. Cluster generated word vectors into concept clusters

	Concept Cluster 1	Concept Cluster 2	Concept Cluster 3
Document 1	3	1	1
Document 2	2	0	2

3. Represent documents as frequencies of each concept (Bag-of-concepts)

Figure 6: Bag-of-Concepts

This paper suggests the bag-of-concepts method as an alternative method for document representation (Figure 6). In this proposed method, word vectors for words in a collection of documents are trained via word2vec. As word2vec embeds semantically similar words into neighboring area, the proposed method clusters neighboring words into one common concept cluster. Similar to the bag-of-words method, each document vector will then be represented by the counts of each concept clusters in the document. As each concept cluster will contain words with similar meaning or common hypernym, the features of the document vectors generated from the proposed method will be interpretable and intuitive. Furthermore, the bag-of-concepts method can be understood as a non-linear dimension reduction technique for transforming a word space

into a concept space based on semantic similarity. As the proposed method represents a document with concept frequencies instead of word frequencies, it incorporate interpretability of the bag-of-words method, and representational superiority and non-sparsity of the distributed representation method, while overcoming their limitations.

As word2vec maximizes the cross product between the embedding vectors and the context vectors, cosine distance metric is used for clustering in the embedding space. Consequently, spherical k-means algorithm [18] is used to cluster word vectors into concept clusters. For predetermined value of k, spherical k-means clustering, similar to k-means clustering, iteratively assigns each data point to one of k centroids and updates each centroid given the membership of the data points. However, spherical k-means clustering, instead of Euclidean distance, uses cosine similarity as a distance metric.

4 Data Set Description

In order to show the representational performance of the proposed method and its applicability, document clustering and classification tasks have been carried out using the document vectors generated from the proposed method. Document clustering task aims at grouping documents according to their correct classes. On the other hand, document classification task generates a model that can distinguish differences between the documents. If the proposed method can truly capture the semantic differences between the documents, it should perform well in these tasks.

Table 1: Reuter dataset labels

Classes	**Number of Documents**
Entertainment	25,500
Sports	25,500
Technology	25,500
Market	25,423
Politics	25,500
Business	25,500
World	25,500
Health	25,500

In this paper, Reuter dataset has been used for these task. To avoid class imbalance problem, Reuter dataset consists of 203,923 randomly selected articles from Reuter website that have been published between September 1st, 2006 and June 6th, 2015. These articles are labeled by Reuter website into 8 different classes as shown in Table 1. The total number of sentences amounts to 3,076,016, while the total number of tokens is equivalent to 89,146,031. For faster word2vec training, we have ignored those words that occurred less than 20 times in the entire dataset,

leaving total of 65,159 unique words to train.

5 Experiment Result

Biggest contribution of proposed bag-of-concepts method is that it incorporates the advantages of the bag-of-words method and doc2vec model. Similar to doc2vec model, the proposed method maintains superior representational performance derived from utilizing contextual information. Furthermore, it creates dense document vectors with reasonable number of dimensions. Yet, the proposed method also provides explicitly explanatory features for the document vectors, providing interpretability for the vectors themselves and explainability for the text mining models built from these vectors. These three aspects of the suggest method – representational performance, vector interpretability, and model explainability – are established through performing document clustering and classification task on Reuter dataset.

5.1 Representation Effectiveness

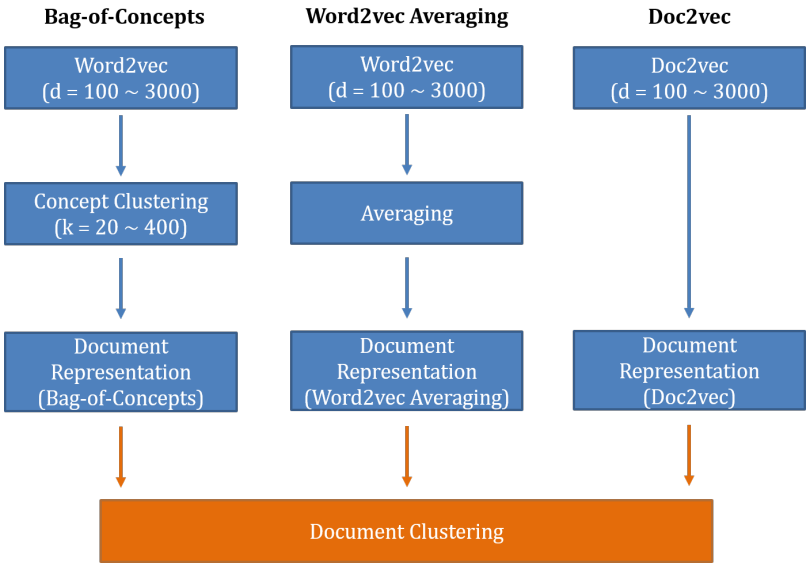


Figure 7: Document clustering experiment design

In order to analyze the representation effectiveness of the proposed method, document clustering task has been carried out on the document vectors generated from the proposed method. Clustering performance is compared to those calculated from the document vectors generated from word2vec averaging method and doc2vec method as shown in Figure 7.

Numerous hyperparameters are involved in training effective word2vec and doc2vec models. In order to minimize the impact of hyperparameters in the overall performance, the proposed method, word2vec averaging method and doc2vec method are designed to share same window size of 9 and training epoch of 3. All word2vec and doc2vec training have been carried out by

using Gensim library[1] in Python. Various number of dimensions for the document and word vectors have been tested. Starting with the dimension of 100, the dimension is increased by 100 until 1000, after which it is increased by 1000 until the dimension of 3000. The proposed method is additionally influenced by an extra hyperparameter k, the number of concept clusters to be constructed. In order to observe its impact on the representation performance, several values for the number of concept clusters have also been tested. Starting with 20, the value of k is increased by 10 until 400.

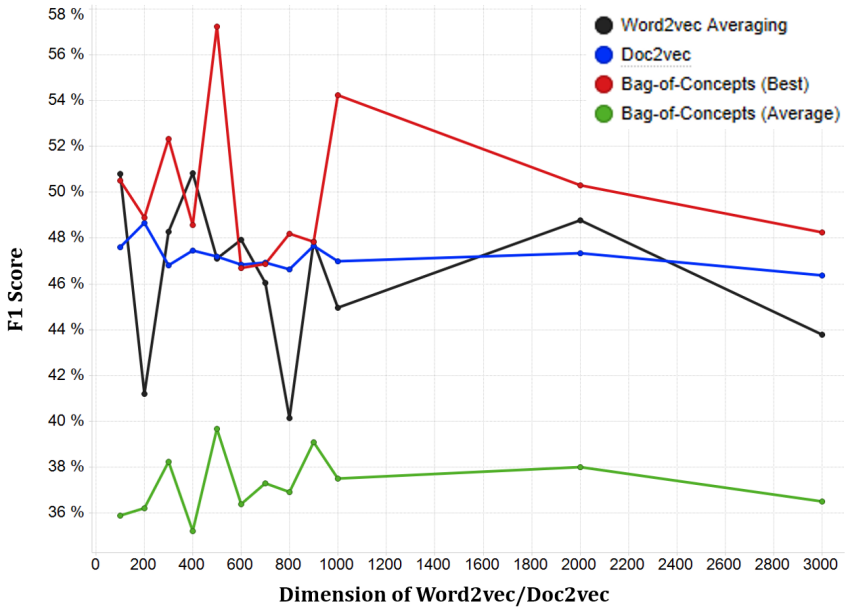


Figure 8: F1 score of document clustering task

Figure 8 and Table 2 show F1 score of the clustering result from these three methods with respect to the number of dimension of word2vec and doc2vec. As shown in the following equation, F1 score uses precision and recall to calculate the accuracy of the test result. As this document clustering task is a case of multiclass clustering, weighted average of F1 scores for each binary case is calculated.

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

For a given dimension, "Bag-of-Concepts (Best)" indicates a model with the highest F1 score amongst all of the models trained with different number of concept clusters, k, between $20 \sim 400$. On the other hand, "Bag-of-Concepts (Average)" represents a model that averages F1 scores from all of the proposed models with different values of k for a given dimension. Full detailed list of F1 score for all values of k is included as an appendix. As shown by Figure 8, average performance of the proposed method is lower than those of doc2vec method and word2vec averaging method. If the number of concept cluster k is carefully selected, the performance of

[1]https://radimrehurek.com/gensim/

the proposed method, however, can drastically improve and is similar or in some hyperparameter setting, can outperform those of current state-of-art doc2vec method and word2vec averaging method. As an alternative document representation method, the proposed method provides an effective representational power at a similar level as doc2vec that have already been shown to outperform the bag-of-words approach [3].

Table 2: List of F1 score for document clustering task

Dimension of Word2vec / Doc2vec	**Bag-of-Concepts (Best)**	**Bag-of-Concepts (Average)**	**Word2vec Averaging**	**Doc2vec**
100	0.505036	0.358903	0.508018	0.47598
200	0.489007	0.362014	0.411944	0.486628
300	0.523417	0.382304	0.482698	0.468047
400	0.485581	0.352057	0.508342	0.474606
500	0.572366	0.396641	0.471125	0.471829
600	0.466831	0.363745	0.479197	0.468465
700	0.468562	0.372937	0.460313	0.469243
800	0.481906	0.369165	0.401519	0.466425
900	0.47838	0.390732	0.478478	0.476587
1000	0.542261	0.37483	0.449473	0.469872
2000	0.502957	0.379956	0.487743	0.473375
3000	0.482591	0.364974	0.437778	0.463631

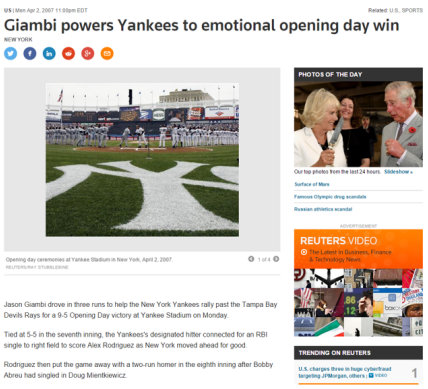
5.2 Vector Interpretability

Unlike doc2vec, the proposed method, while still maintaining the representational effectiveness of doc2vec, is capable of providing intuitive interpretation for the generated document vectors. In order to show this vector interpretability, we will use the proposed method with the highest F1 score. In this model, all of the words are embedded into continuous space of 500 dimensions, and are clustered into 110 concept clusters (k = 110). Furthermore, two clearly different documents are selected as examples as shown in Figure 9.

In Figure 9, Document 1 belongs to Sports class as it discusses about an opening day win for New York Yankees, a baseball team. Document 2, on the other hand, belongs to Politics class as it discusses a recent survey regarding the Trans-Pacific Partnership agreement, a economic trade agreement between twelve countries around Pacific Rim. Both doc2vec and the proposed method successfully cluster Document 1 as a member of Sports class, while Document 2 as a member of Politics class. Observing the document vectors generated from the proposed method, however, provides more insightful and profound understanding behind the result. The features

Features	X[0]	...	X[33]	...	X[108]	X[109]
Document 1	5	...	1	...	0	0
Document 2	27	...	36	...	1	0

[Document 1]



[Document 2]

Figure 9: Examples of interpretable document vectors

of the document vectors generated from doc2vec represent the coordinates of the vectors in 500 dimensional space, but it fails to provide any clear intuitive understanding behind the meaning of each axis. The proposed method, however, successfully offers clear interpretation of the meaning behind each features, explaining each document's clustering result through the concept clusters.

Word	Distance to Centroid
Astros	0.209113
Playoff-bound	0.216279
Phillies	0.231677
Last-place	0.232075
Timberwolves	0.237807
Mariners	0.242180
Flyers	0.245423
Thrashers	0.247595
Sabres	0.250336
Devils	0.252015
Blackhawks	0.255871
Orioles	0.256698
Athletics	0.260109

- **Sport Teams**
- Concept Frequency:
  Doc 1: 14 vs. Doc 2: 0

Word	Distance to Centroid
Fourth-inning	0.188195
Aybar	0.201127
Pinch-hit	0.217082
Pinch-hitter	0.221174
Hitless	0.227714
First-inning	0.236647
DH	0.240897
Two-out	0.241593
Okajima	0.249996
No-hit	0.250199
Delmon	0.253375
Kozma	0.255309
Eighth-inning	0.255412

- **Baseball Terminologies**
- Concept Frequency:
  Doc 1: 68 vs. Doc 2: 1

Figure 10: Concept clusters that are strongly related to Document 1

Figure 10 ∼ 13 list some examples of contrasting features in two document vectors generated from the proposed method that provide some intuition behind the clustering result. Looking at the words in the concept clusters depicted in Figure 10, we can understand that these two concept clusters contain words that are related to the names of sports teams, and to baseball terminologies respectively. In Document 1, words belonging to the concept cluster related to the names of sports teams occurred 14 times compared to none in Document 2. Similarly, the concept cluster related to baseball terminologies occurred 68 times in Document 1, while once in Document 2. Consequently, we can understand that Document 1 contains more words related to the names of sports teams and to baseball terminologies. As Document 1 is indeed an article about a baseball game, it seems inevitable for Document 1 to have high occurrences in these two concept clusters. As these concepts are more likely to be used in a sports section of a newspaper than a politics section, Document 1, therefore, is clustered into Sports class, while Document 2 isn't.

Word	Distance to Centroid
Fretilin	0.298141
Hard-left	0.299046
Smer	0.300370
Ovp	0.300925
Greens	0.303287
Socialists	0.305534
Party	0.310117
Peronist	0.321366
Kke	0.324051
Pis	0.333701
Congress-led	0.336214
Centrists	0.340830
Pro-eu	0.343883

- **Political Party**
- Concept Frequency: **Doc 1: 5 vs. Doc 2: 27**

Word	Distance to Centroid
Six-nation	0.341851
Negotiations	0.358357
Final-status	0.358551
Talks	0.369950
Accord	0.384951
Two-track	0.388305
Agreement	0.388699
Working-level	0.401054
Long-stalled	0.411923
Trilateral	0.416301
Deal	0.417467
Disarmament	0.423539
Israeli-Syrian	0.424372

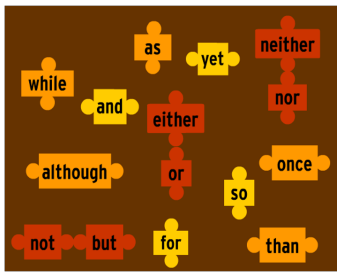
- **Negotiation & Treaty**
- Concept Frequency: **Doc 1: 1 vs. Doc 2: 36**

Figure 11: Concept clusters that are strongly related to Document 2

Looking at the words in the concept clusters depicted in Figure 11, we can understand that these two concept clusters contain words that are related to the names of political parties, and to the words that describe negotiations respectively. In Document 2, words belonging to the concept cluster related to the names of political parties occurred 27 times compared to 5 times in Document 1. Similarly, the concept cluster related to the negotiation terms occurred 36 times

in Document 2, while once in Document 1. Consequently, we can understand that Document 2 contains more words related to the names of political parties and to the concept of negotiation. As these concepts are more likely to be used in a political section of a newspaper than a sports section, Document 2, therefore, is clustered into Politics class, while Document 1 isn't clustered into the same class.

Word	Distance to Centroid
While	0.378267
But	0.384359
However	0.387299
Although	0.388328
Only	0.417179
Now	0.421535
Then	0.424409
Also	0.425922
Another	0.439093
The	0.449224
May	0.449749
Leaving	0.451124
That	0.451503



- **Conjunctions**

- Concept Frequency:
Doc 1: 146 vs. Doc 2: 198

Figure 12: Concept clusters that are strongly related to both documents

First four concept clusters in Figure 10 $\sim$ 11 successfully capture the contents of documents and provide reasons behind why each document is clustered into Sports and Politics classes respectively. However, not every concept clusters are effective in providing intuition behind the clustering result. Figure 12 shows a concept cluster that occurred most frequently in both Document 1 and 2. Looking at some of the words within this concept cluster, it becomes obvious that conjunctions are clustered into this concept cluster. As conjunctions can be common in any articles, the occurrences of this concept cluster in both documents are relatively higher compared to the occurrences of other concept clusters. Thus, this concept cluster, despite its high occurrence, is irrelevant in capturing meaningful differences of these two documents.

Word	Distance to Centroid
Sirnak	0.190246
Barzeh	0.216446
Qaboun	0.218347
Sidon	0.218943
Mukalla	0.226163
Mosul	0.231129
Hama	0.232689
Adhamiya	0.233669
Ramadi	0.235161
Jobar	0.241562
Vabroud	0.242106
Kerbala	0.242618
Gunbattles	0.243645

- **Middle Eastern Cities**

- Concept Frequency:
Doc 1: 37 vs. Doc 2: 11

Figure 13: Misallocated concept clusters

The concept cluster in Figure 13 represent the names of Middle Eastern cities. Although both Document 1 and 2 don't contain any words related to the cities in the Middle East, the

occurrences of this concept cluster in these two documents are quite significant. Through careful observation of the words in this concept cluster, it can be discovered that such high frequency of this irrelevant concept cluster has occurred due to misallocation of some irrelevant terms into this concept cluster. For example, some common words such as "near" and "cities" have been clustered into this concept cluster. Consequently, occurrence of such irrelevant yet common words in the documents has increased the frequency of the corresponding concept cluster in these document vectors without revealing their contrasting contents.

Although some of the concept clusters with high frequencies are not so intuitive in distinguishing these two document vectors, the proposed method, unlike doc2vec, is capable of providing clear interpretation behind the features of the generated document vectors. Through this vector interpretability, it is now possible to understand the comprising contents of the documents, and to comprehend the similarities and the differences between the documents.

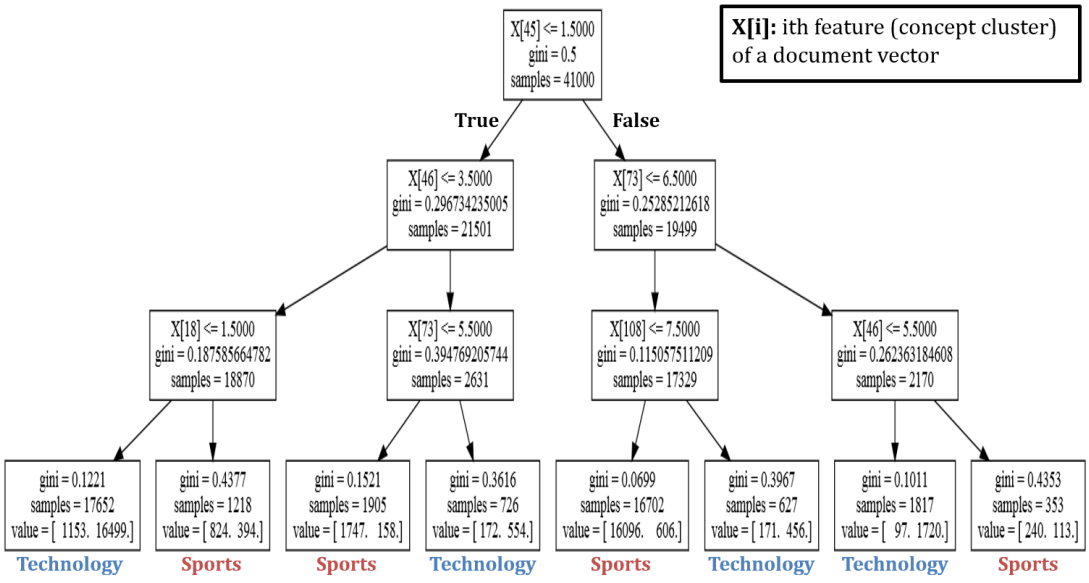
5.3 Model Explainability

The proposed method can additionally provide explanatory power for a text mining model built from the generated document vectors. In order to show such model explainability, a document classifier using decision tree algorithm has been constructed to classify articles in Sports class from those in Technology class. For this decision tree, document vectors are represented by the bag-of-concepts method with the highest clustering performance (dimension of word2vec = 500, k = 110). Amongst 110 concept clusters, this decision tree seeks to identify important concept clusters that can distinguish between Sports and Technology class. Amongst 25,500 articles for each class, 20,500 articles from each class (total of 51,000) have been used to build a decision tree, while remaining 5,000 articles from each class (total of 10,000) have been used as a test set (Table 3). All of the document vectors are represented by the proposed method. The constructed decision tree and its training and test accuracy are shown in Figure 14.

Class	**Total Number of Documents**	**Training Set**	**Test Set**
Sports	25,500	20,500	5,000
Technology	25,500	20,500	5,000

Table 3: Training set and test set for decision tree

Unlike a decision tree generated from doc2vec vectors, this generated decision tree provides an intuitive explanation behind the tree. As each node of the tree represents a specific concept cluster, we can understand the operating logic and the intrinsic characteristics of the classifier and the dataset.



Rule 1: IF (X[45] $\leq$ 1.5) AND (X[46] $\leq$ 3.5) AND (X[18] $\leq$ 1.5) THEN (Y = Technology)
Rule 2: IF (X[45] $\leq$ 1.5) AND (X[46] $\leq$ 3.5) AND (X[18] $>$ 1.5) THEN (Y = Sports)
Rule 3: IF (X[45] $\leq$ 1.5) AND (X[46] $>$ 3.5) AND (X[73] $\leq$ 5.5) THEN (Y = Sports)
Rule 4: IF (X[45] $\leq$ 1.5) AND (X[46] $\leq$ 3.5) AND (X[73] $>$ 5.5) THEN (Y = Technology)
Rule 5: IF (X[45] $>$ 1.5) AND (X[73] $\leq$ 6.5) AND (X[108] $\leq$ 7.5) THEN (Y = Sports)
Rule 6: IF (X[45] $>$ 1.5) AND (X[73] $\leq$ 6.5) AND (X[108] $>$ 7.5) THEN (Y = Technology)
Rule 7: IF (X[45] $>$ 1.5) AND (X[73] $>$ 6.5) AND (X[46] $\leq$ 5.5) THEN (Y = Technology)
Rule 8: IF (X[45] $>$ 1.5) AND (X[73] $>$ 6.5) AND (X[46] $>$ 5.5) THEN (Y = Sports)

Figure 14: Constructed decision tree

Figure 15 lists some concept clusters that the decision tree uses to classify Sports class from Technology class. First splitting node (root) occurs at 45th concept cluster of the document vectors. Top 10 words in this cluster that are closest to the centroid seem to indicate that this concept cluster contains the names of people. Through exploring Reuter website, we have discovered that these words are indeed the names of the reporters, who mainly write sports articles. Consequently, it becomes evident that this classifier considers the names of the reporters as an important criteria for differentiating two classes. Next, we will look at the splitting nodes prior to the leaf nodes. Looking at the left most splitting node, we find that if the value in the 18th feature of a document vector is less than 1.5, corresponding document belongs to Technology class, while if it is bigger than 1.5, it belongs to Sports class. This decision rule becomes intuitively clear if we look at the concept cluster that this feature represents. From Figure 15, we can identify that the terms strongly related to golf scores are clustered into this concept cluster. This node, consequently, classify the documents according to the occurrences of the golf scoring terms. Looking at the actual headlines of the documents that are being classified at this node (Figure 16), we indeed see that this node successfully manages to classify golf articles from other articles.

X[45]: Strongly related to the names of sports new reporters

Word	Distance to Centroid
Himmer	0.14547
Chadband	0.14665
Mehaffey	0.16566
Cambers	0.17331
Manuele	0.17460
Collings	0.18333
Rogovitskiy	0.19248
Thomazeau	0.19778
Vignal	0.19787
Fylan	0.20605

X[46] = Strongly related to sports honors and associations

Word	Distance to Centroid
drawcards	0.34262
over-age	0.41479
multi-sports	0.43338
multi-sport	0.44926
1908	0.46296
honours	0.46650
cups	0.47149
fourth-best	0.47747
WTAs	0.48097
player	0.48181

X[18]: Strongly related to golf scoring terms

Word	Distance to Centroid
back-nine	0.23369
double-bogeys	0.23978
eagling	0.24029
congressional	0.24441
six-over	0.24894
five-over	0.24914
seven-over	0.25737
one-over	0.25855
five-birdie	0.26099
three-putting	0.26230

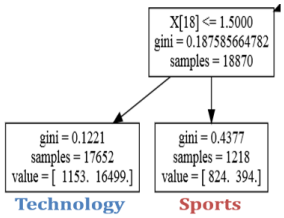
X[73]: Strongly related to descriptions of computer software and internet service

Word	Distance to Centroid
web-surfing	0.30672
apps	0.34588
bandwidth-hungry	0.35815
software-based	0.35873
datacenters	0.36870
data-heavy	0.36984
satellite-based	0.37612
customizing	0.37953
full-featured	0.37958
voice-recognition	0.38839

X[108]: Strongly related to names of online platforms or communities

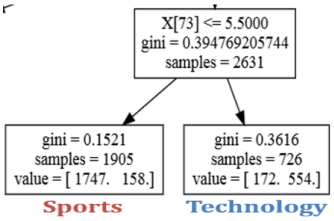
Word	Distance to Centroid
photobucket	0.30672
adsense	0.34588
taobao.	0.35815
mog	0.35873
google+	0.36870
spotify	0.36984
vudu	0.37612
hulu	0.37953
wordpress	0.37958
iqiyi	0.38839

Figure 15: Concept clusters of each nodes



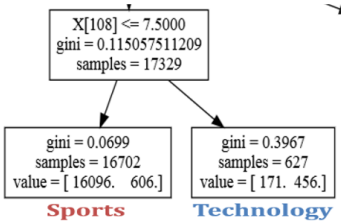
Headline from Technology Node
Spending on video downloads to surge: study
Senator Schumer asks FTC to probe Apple, Android
Konica and GE to jointly develop OLED lights
Get up! No stalling! Virtual life coach is calling
Motorola to buy video technology supplier Terayon

Headline from Sports Node
Johnnie Walker Classic to switch to S. Korea: report
Durant eases to four-stroke victory at Disney
Harrington tops list as Singh wins Volvo Masters
Scott moves three shots clear after lucky eagle
“Scary” Tiger back in contention with Shanghai 64



Headline from Sports Node
Friends La Russa and Leyland in opposite dugouts
Pacquiao, Morales primed despite lack of world title
On Rio's beaches, Olympic excitement and doubts
Rio making strong progress for 2016 Olympics: IOC
1985 Chicago Bears get long overdue White House welcome

Headline from Technology Node
Hit streaming service Spotify eyes U.S. music fans
Apple takes on Google with own maps, better Siri
Star Wars-inspired prototype creates holographic dis play
Newsmaker: Nokia's Elop eyes Microsoft window of opportunity
Apple's Jobs takes stage to talk iCloud



Headline from Sports Node
Choi keeps Chrysler lead with late flourish
Smith strike lifts Oilers to revenge victory over Hurricanes
Sampras says he could have held his own against Federer
Dutch must stay focused on main goal, says Sneijder
Wounded Warriors rally to stun Celtics

Headline from Technology Node
Amazon's Kindle reader breaks monthly sales record
Hacking "mole" helps FBI arrest Anonymous leaders
Alibaba in funding talks with India's Snapdeal: source
Baidu, China sued in U.S. for Internet censorship
LinkedIn site disrupted in protest-wary China

Figure 16: Headlines of documents in each leaf node

Similar results follow for other remaining three nodes. Through these nodes, we can understand that golf and sports associations are two major concept clusters that this classifier uses to differentiate the documents in Sport class from those in Technology class. Similarly, we realize that computer software related terms and the names of online platforms are two major concept clusters that this classifier utilizes for differentiating the documents in Technology class from those in Sports class. As Figure 16 shows, the headlines of the articles that are distinguished at these nodes further substantiate the importance of these concept clusters in the classifier as they appear to be more relevant to their corresponding concept clusters. Although similar classification task can be carried out by doc2vec, it cannot provide intuitive explanation behind the operating logic of the classifier unlike the proposed method as shown by this example.

6 Conclusion

This paper proposes the bag-of-concepts method for representing a document vector, through which the advantages of the bag-of-words method and doc2vec are integrated to overcome the weaknesses of each model. While preserving semantic similarity of the word vectors, the proposed method clusters the word vectors generated from word2vec into concept clusters. Consequently, the proposed method maintains the representational effectiveness and non-sparsity of doc2vec, while providing intuitive vector interpretability at the same time. With intuitive vector interpretability, we can acquire more explicit and profound understanding of the document vectors and their differences. If the proposed method is applied in specific text mining task such as document classification task, we can furthermore comprehend the operating logic and unique characteristics behind the built model. Consequently, even those who aren't experts in text mining and data mining can easily understand and accept the constructed model and its constituting vectors. Due to these vector interpretability and model explainability, the proposed method can be applied in solving various real business problems, in which document representation itself is not the only issue.

In this paper, the labels of the concept clusters have been manually determined. In future works, however, we will explore ways to label the concept clusters semi-automatically or automatically, providing more objective labels for describing the concept clusters. Furthermore, we will also compare the impacts of various clustering algorithm in the quality of the generated concept clusters. With further exploration, we hope that the proposed method will establish itself as a fundamental building block for solving various text mining problems arising from real business problems.

7 Acknowledgement

This work was supported by the BK21 Plus Program(Center for Sustainable and Innovative Industrial Systems, Dept. of Industrial Engineering, Seoul National University) funded by the Ministry of Education, Korea (No. 21A20130012638), the National Research Foundation(NRF) grant funded by the Korea government(MSIP) (No. 2011-0030814), and the Institute for Industrial Systems Innovation of SNU.

8 Appendix

This appendix includes a complete list of F1 score of clustering accuracy resulting from all values of k (the number of concept clusters) used for clustering words into concept clusters. For each word2vec embedding dimension between $100 \sim 3000$, the proposed bag-of-concepts method has used varying number of concept clusters between $20 \sim 400$ to create document vectors for document clustering task. Following F1 score lists weighted average of F1 scores of the clustering result. As this document clustering task is a case of multiclass clustering, weighted average of F1 scores for each binary case is calculated.

<Word2vec Dimension: 100>

Number of Concept Clusters	F1 Score
20	0.415555
30	0.474610
40	0.378772
50	0.410554
60	0.328951
70	0.376325
80	0.414693
90	0.349167
100	0.333124
110	0.437022
120	0.365336
130	0.448871
140	0.325328

Number of Concept Clusters	F1 Score
150	0.393033
160	0.261714
170	0.505036
180	0.319439
190	0.304670
200	0.454799
210	0.328831
220	0.278632
230	0.341028
240	0.288940
250	0.312114
260	0.403633
270	0.273404

Number of Concept Clusters	F1 Score
280	0.270626
290	0.371215
300	0.423299
310	0.269386
320	0.377150
330	0.287306
340	0.280474
350	0.317728
360	0.370874
370	0.388643
380	0.422034
390	0.331185
400	0.363711

<Word2vec Dimension: 200>

Number of Concept Clusters	F1 Score
20	0.427086
30	0.431358
40	0.285255
50	0.389196
60	0.351606
70	0.384257
80	0.383991
90	0.391219
100	0.347378
110	0.386250
120	0.299218
130	0.405387
140	0.451887

Number of Concept Clusters	F1 Score
150	0.290079
160	0.439531
170	0.448563
180	0.443253
190	0.339252
200	0.415949
210	0.489007
220	0.304512
230	0.403788
240	0.290145
250	0.400820
260	0.259588
270	0.338136

Number of Concept Clusters	F1 Score
280	0.303496
290	0.397334
300	0.285698
310	0.416158
320	0.328379
330	0.347461
340	0.372715
350	0.385729
360	0.245000
370	0.283378
380	0.382478
390	0.235845
400	0.338156

<Word2vec Dimension: 300>

Number of Concept Clusters	F1 Score
20	0.469121
30	0.396240
40	0.372656
50	0.461167
60	0.390214
70	0.340326
80	0.294913
90	0.317215
100	0.343798
110	0.362368
120	0.273419
130	0.395809
140	0.352353

Number of Concept Clusters	F1 Score
150	0.328402
160	0.490790
170	0.409920
180	0.423966
190	0.317032
200	0.523417
210	0.310588
220	0.302058
230	0.352011
240	0.461796
250	0.445265
260	0.282231
270	0.356177

Number of Concept Clusters	F1 Score
280	0.413613
290	0.420020
300	0.373074
310	0.368674
320	0.341753
330	0.294131
340	0.403327
350	0.438766
360	0.397131
370	0.444468
380	0.377441
390	0.458492
400	0.405720

<Word2vec Dimension: 400>

Number of Concept Clusters	F1 Score
20	0.333606
30	0.351357
40	0.405270
50	0.337873
60	0.409927
70	0.271142
80	0.292622
90	0.297865
100	0.485581
110	0.332732
120	0.351875
130	0.393674
140	0.320539

Number of Concept Clusters	F1 Score
150	0.371875
160	0.483382
170	0.409580
180	0.356964
190	0.307050
200	0.240380
210	0.349123
220	0.419743
230	0.400156
240	0.390202
250	0.398172
260	0.284326
270	0.258879

Number of Concept Clusters	F1 Score
280	0.352280
290	0.384273
300	0.277114
310	0.362391
320	0.422079
330	0.212798
340	0.384408
350	0.366316
360	0.483013
370	0.333068
380	0.253057
390	0.346025
400	0.299500

<Word2vec Dimension: 500>

Number of Concept Clusters	F1 Score
20	0.395951
30	0.391638
40	0.411754
50	0.378102
60	0.363853
70	0.352856
80	0.377581
90	0.405359
100	0.417798
110	0.572366
120	0.404436
130	0.463939
140	0.246201

Number of Concept Clusters	F1 Score
150	0.339878
160	0.427921
170	0.377913
180	0.415552
190	0.348664
200	0.443124
210	0.364274
220	0.409479
230	0.396445
240	0.344521
250	0.446078
260	0.353732
270	0.340669

Number of Concept Clusters	F1 Score
280	0.419234
290	0.463710
300	0.450911
310	0.388924
320	0.496007
330	0.315670
340	0.350163
350	0.369696
360	0.337418
370	0.321418
380	0.464495
390	0.422274
400	0.264777

<Word2vec Dimension: 600>

Number of Concept Clusters	F1 Score
20	0.402129
30	0.398887
40	0.350267
50	0.381521
60	0.362067
70	0.315019
80	0.375195
90	0.408715
100	0.353451
110	0.390290
120	0.378848
130	0.299130
140	0.402655

Number of Concept Clusters	F1 Score
150	0.357346
160	0.371573
170	0.407402
180	0.315076
190	0.417321
200	0.339547
210	0.362933
220	0.243342
230	0.466831
240	0.448229
250	0.373031
260	0.284453
270	0.400910

Number of Concept Clusters	F1 Score
280	0.358365
290	0.268001
300	0.334110
310	0.305035
320	0.354579
330	0.384368
340	0.443050
350	0.344798
360	0.347933
370	0.325332
380	0.338222
390	0.384991
400	0.391113

<Word2vec Dimension: 700>

Number of Concept Clusters	F1 Score
20	0.366522
30	0.431582
40	0.340148
50	0.328871
60	0.270981
70	0.392649
80	0.309403
90	0.39507
100	0.352024
110	0.348395
120	0.335724
130	0.385444
140	0.288624

Number of Concept Clusters	F1 Score
150	0.406182
160	0.449741
170	0.36062
180	0.38309
190	0.391135
200	0.352237
210	0.373929
220	0.315777
230	0.353668
240	0.468563
250	0.328499
260	0.438187
270	0.415148

Number of Concept Clusters	F1 Score
280	0.447993
290	0.250474
300	0.362702
310	0.42902
320	0.346669
330	0.443061
340	0.330436
350	0.451433
360	0.381593
370	0.458017
380	0.33816
390	0.339703
400	0.383084

<Word2vec Dimension: 800>

Number of Concept Clusters	F1 Score
20	0.401721
30	0.380336
40	0.406026
50	0.380430
60	0.441462
70	0.356188
80	0.392441
90	0.374856
100	0.240308
110	0.405297
120	0.399190
130	0.481906
140	0.417466

Number of Concept Clusters	F1 Score
150	0.336288
160	0.407846
170	0.330830
180	0.332509
190	0.354566
200	0.446010
210	0.369566
220	0.365387
230	0.348978
240	0.371595
250	0.387117
260	0.478853
270	0.429617

Number of Concept Clusters	F1 Score
280	0.435919
290	0.302491
300	0.374972
310	0.437256
320	0.379242
330	0.295973
340	0.233079
350	0.346914
360	0.218122
370	0.463968
380	0.288570
390	0.320174
400	0.263949

<Word2vec Dimension: 900>

Number of Concept Clusters	F1 Score
20	0.385082
30	0.354805
40	0.400793
50	0.423003
60	0.424248
70	0.408771
80	0.383305
90	0.390473
100	0.407109
110	0.410944
120	0.377461
130	0.376412
140	0.371279

Number of Concept Clusters	F1 Score
150	0.289581
160	0.276944
170	0.348645
180	0.457593
190	0.347253
200	0.401210
210	0.456976
220	0.394590
230	0.377922
240	0.408241
250	0.405916
260	0.455534
270	0.335838

Number of Concept Clusters	F1 Score
280	0.447598
290	0.418377
300	0.365726
310	0.358911
320	0.259511
330	0.446463
340	0.393850
350	0.478380
360	0.457273
370	0.289118
380	0.430024
390	0.409347
400	0.414035

<Word2vec Dimension: 1000>

Number of Concept Clusters	F1 Score
20	0.396292
30	0.441013
40	0.397271
50	0.301283
60	0.384815
70	0.277849
80	0.361537
90	0.382136
100	0.401325
110	0.430278
120	0.352961
130	0.302100
140	0.491752

Number of Concept Clusters	F1 Score
150	0.425036
160	0.429285
170	0.298406
180	0.463932
190	0.381341
200	0.472851
210	0.458013
220	0.376800
230	0.389287
240	0.313312
250	0.419160
260	0.382872
270	0.340604

Number of Concept Clusters	F1 Score
280	0.251288
290	0.274291
300	0.280222
310	0.349719
320	0.419679
330	0.257833
340	0.379167
350	0.416273
360	0.307873
370	0.241681
380	0.542261
390	0.475590
400	0.350964

<Word2vec Dimension: 2000>

Number of Concept Clusters	F1 Score
20	0.440328
30	0.421134
40	0.404517
50	0.379935
60	0.323401
70	0.458171
80	0.356961
90	0.399426
100	0.448388
110	0.386497
120	0.337120
130	0.435419
140	0.377756

Number of Concept Clusters	F1 Score
150	0.340930
160	0.406940
170	0.396413
180	0.447934
190	0.358917
200	0.392061
210	0.399540
220	0.330953
230	0.437065
240	0.449841
250	0.382267
260	0.320113
270	0.502957

Number of Concept Clusters	F1 Score
280	0.378311
290	0.251787
300	0.491422
310	0.266930
320	0.300174
330	0.280962
340	0.445168
350	0.247429
360	0.303836
370	0.388624
380	0.441073
390	0.252259
400	0.435329

<Word2vec Dimension: 3000>

Number of Concept Clusters	F1 Score
20	0.427639
30	0.367846
40	0.315287
50	0.306316
60	0.297440
70	0.423893
80	0.436830
90	0.283412
100	0.293954
110	0.262525
120	0.372138
130	0.303183
140	0.428180

Number of Concept Clusters	F1 Score
150	0.361219
160	0.370973
170	0.305327
180	0.377924
190	0.267442
200	0.387804
210	0.482591
220	0.454687
230	0.365219
240	0.436401
250	0.442568
260	0.381583
270	0.391629

Number of Concept Clusters	F1 Score
280	0.386741
290	0.449469
300	0.457489
310	0.361214
320	0.416924
330	0.355359
340	0.332080
350	0.457322
360	0.277339
370	0.376619
380	0.261301
390	0.220435
400	0.337675

References

[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval*. ACM Press, 1999.

[2] Mohi Bansal, Kevin Gimpel, and Karen Livescu. Tailoring continuous word representations for dependency parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2014.

[3] Andrew M Dai, Christopher Olah, Quoc V Le, and Greg S Corrado. Document embedding with paragraph vectors. In *NIPS Deep Learning Workshop*, 2014.

[4] Scott C Deerwester, Susan T Dumais, Thomas K Landauer, George W Furnas, and Richard A Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):1986–1998, 1990.

[5] Zellig S Harris. Distributional structure. *Word*, 10:146–162, 1954.

[6] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[7] Anna Huang. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference NZCSRSC2008*, pages 49–56, 2008.

[8] Pranam Kolari, Akshay Java, Tim Finin, Tim Oates, and Anupam Joshi. Detecting spam blogs: A machine learning approach. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1351–1356. Association for the Advancement of Artificial Intelligence, 2006.

[9] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv*, 1405.4053(2), 2014.

[10] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv*, 1301.3781(3), 2013.

[12] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv*, 1309.4168(1), 2013.

[13] Mengye Ren, Ryan Kiros, and Richard S Zemel. Exploring models and data for image question answering. In *Advances in Neural Information Processing Systems*, 2015.

[14] Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

[15] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[16] Lei Wu, Steven CH Hoi, and Nenghai Yu. Semantics-preserving bag-of-words models and applications. *Image Processing, IEEE Transactions on*, 19(7):1908–1920, 2010.

[17] Chao Xing, Dong Wang, Xuewei Zhang, and Liu Chao. Document classification with distributions of word vectors. In *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*, 2014.

[18] Shi Zhong. Efficient online spherical k-means clustering. In *Proceedings of International Joint Conference on Neural Networks*, 2005.