International Journal of Data Science and Analytics (2020) 10:49–64
https://doi.org/10.1007/s41060-019-00200-5

REGULAR PAPER

# A document representation framework with interpretable features using pre-trained word embeddings

Narendra Babu Unnam[1] · P. Krishna Reddy[1]

Received: 8 February 2019 / Accepted: 13 November 2019 / Published online: 25 November 2019

**Abstract**

We propose an improved framework for document representation using word embeddings. The existing models represent the document as a position vector in the same word embedding space. As a result, they are unable to capture the multiple aspects as well as the broad context in the document. Also, due to their low representational power, existing approaches perform poorly at document classification. Furthermore, the document vectors obtained using such methods have uninterpretable features. In this paper, we propose an improved document representation framework which captures multiple aspects of the document with interpretable features. In this framework, a document is represented in a different feature space by representing each dimension with a potential feature word with relatively high discriminating power. A given document is modeled as the distances between the feature words and the document. To represent a document, we have proposed two criteria for the selection of potential feature words and a distance function to measure the distance between the feature word and the document. Experimental results on multiple datasets show that the proposed model consistently performs better at document classification over the baseline methods. The proposed approach is simple and represents the document with interpretable word features. Overall, the proposed model provides an alternative framework to represent the larger text units with word embeddings and provides the scope to develop new approaches to improve the performance of document representation and its applications.

**Keywords** Text mining · Feature engineering · Document representation · Document classification · Word embeddings

## 1 Introduction

The distributional hypothesis [34] states that the words which occur in similar contexts will have a similar meaning. Operationalizing this theory, a family of neural language models [26,28,33] have been proposed to represent the words as position vectors or *word embeddings (WEs)* in low-dimensional feature spaces. The WEs are the result of internal weights of the neural network models which implicitly learn about the words co-occurrence counts information [6,39]. The intrinsic nature of these WEs is that the words which are contextually similar are closer in the corresponding feature space. The recent WE models [6,26,28,33,39] are typically trained on huge text corpora. So the resultant vectors encode many linguistic regularities [30] along with solid syntactic and semantic information. Due to the rich information encoded in these WEs, they are used in multiple downstream tasks such as dependency parsing, named entity recognition, part of speech tagging, text clustering, and text classification.

In this paper, we address the issue of improving the document representation by exploiting WEs. The most commonly used composition function to represent the given document is *vector averaging of the corresponding WEs*. An approach has been proposed in [1] to represent the document with the positional vector, which is obtained by weighted averaging of the WEs of the document after assigning weights to the words. A method was proposed in [25], popularly known as *Doc2vec*, by adopting *Word2vec* for documents. The *Doc2vec* model treats each document as another context word and co-learns the embeddings for both words and documents.

The models based on averaging [1,25] represent the document in the same feature space as that of WEs, i.e., essentially treating the document as another word in the WE space. So

✉ Narendra Babu Unnam
narendra.unnam@research.iiit.ac.in

P. Krishna Reddy
pkreddy@iiit.ac.in

[1] Kohli Centre on Intelligent Systems, IIIT Hyderabad, Hyderabad, India

they suffer from the natural drawbacks of averaging. Firstly, the existing approaches oversimplify the document representation by representing the document as a position vector in the same WE space. Secondly, two different distributions in a feature space can have the same average. As a result, two documents conversing about different topics can end-up with fairly close vector representations. So the existing models based on averaging are unable to capture the multiple aspects as well as the broad context in the document.

Semantic interpretability of features is a desirable property [13] of a representation model which enables the humans to understand, develop insights and explain the features. Word embeddings are weights from hidden layers of neural language models, so the semantic space in which they are represented often have the latent features/dimensions. The semantic meaning encoded in each feature/dimension of an individual word embedding is unclear and uninterpretable. Consequently, document representations that are derived by compositing the word embeddings component-wise [1,25] are also limited by the inherent uninterpretable features.

Instead of representing the document as another word in WE space, there is a scope to represent the document in terms of distances between the fixed number of potential interpretable feature words to the document using WEs. With this, it is possible to capture the multiple aspects of the document in a comprehensive manner. In other words, it is possible to form a composition function to represent the document by considering a different feature space instead of the WE space and use the words as the features of the document representation rather than with the latent features (dimensions) of WE space.

Given the dataset of documents and pre-trained WEs, in this paper, we present a novel framework to represent the documents as fixed-length feature vectors. In the proposed framework, which we call as (DIFW) document representation with interpretable features using WEs framework, each dimension in document feature vector corresponds to a potential feature word with relatively high discriminating power, and the given document is represented as its distances from the potential feature words.

The DIFW framework is simple and represents the document by capturing the multiple topics in the document in an effective manner. Also, it represents the document with interpretable word features.

The key contributions of this work are fourfold:

1. Given the dataset of documents and pre-trained WEs, we present DIFW framework to represent the given document.
2. As a part of the DIFW framework, we propose two criteria for the selection of potential feature words based on the frequency distribution and spatial distribution of words.
3. As a part of the DIFW framework, we discussed multiple distance definitions from the literature and proposed a new distance measure to find the distance between the given word and document.
4. We have conducted extensive experiments on multiple datasets to demonstrate the utility of the DIFW framework. Experimental results show that the proposed framework consistently performs better at document classification over the baseline methods.

The remainder of this paper is organized as follows. In Sect. 2, we discuss related works. In Sect. 3, we present the proposed framework. In Sect. 4, we explain the experimental results. Finally, we conclude in Sect. 5 with directions for future work.

## 2 Related work

In this section, we discuss the word embedding models and related composition functions to represent the larger text units such as sentences and documents using word embeddings.

Word embeddings are the internal weight vectors in language modeling architectures which are used to represent the words. The idea of using neural networks to learn the distributed representations for words is introduced in [3]. The model proposed in [3] learns the representations as part of a neural network architecture for language modeling. The word representations are learned along with the parameters of the language model. In [8,9], a model was proposed to learn the general word representations instead of task-specific representations. The model learns the word representations by training on vast amounts of text corpora and the learned representations are used for downstream NLP tasks.

These models are computationally expensive due to the hidden layers in their architectures. A computationally efficient model called Word2vec model was proposed in [28,29] to learn the word representations by predicting the neighboring words for any given word. A count-based model named GloVE was proposed in [33], in which global words co-occurrence statistics are used in order to produce word vectors. With the success of these WE models, there have been continuous works [6,7,16,19,26,31,39] to improve the quality of WEs.

Research efforts are being made to develop the composition functions which take word representations as input and produce representations of document. Most commonly used composition function is vector averaging. In this method, component-wise mean across all the WEs corresponding to all the words in a document is calculated and the resultant vector is used as the document representation. In [1], authors introduce a word weighting method named smooth inverse frequency. The weight of a word $w$ is calculated as $a/(a + p(w))$, where $a$ is the parameter and $p(w)$ is the frequency of the word $w$. In this method, first, weighted averages of the word embeddings are calculated and then projections of the average vectors on their first principal component are removed.

In [25], Doc2vec, an extension of Word2vec, is proposed to generate feature vectors for documents. In this model, each document is treated as an extra neighboring word and its vector representation is co-learned along with the words representations. In [20,40], all the words in the vocabulary are clustered by applying $k$-means algorithm over the word vectors. The resultant clusters of words represent the concepts, and these concepts are used as features for document representation. The associative strength between a document and a cluster is computed as the number of words that exist in both.

Skip-thought vectors [22] is an adaption of Word2vec for sentence representation. In this method, the sentences are the basic units rather than words. For a given sentence, an encoder-decoder model is used to reconstruct its surrounding sentences. Similar to word vector averaging, feature vectors of sentences in a document can be averaged to represent the document.

The proposed framework in this paper is different from preceding approaches in the following manner. The proposed framework represents the documents in a higher-dimensional space rather than the word embedding space. Word embeddings capture the meanings of words as the spatial distances among them. So, in the proposed framework, we model a document as set of distances rather than as the latent features.

## 3 Proposed framework

### 3.1 Proposed model

Notably, the existing document representation approaches based on averaging essentially model the document as another word in the given WE space. Since WEs have latent features, the document representation which is the mean of WEs also inherits these latent features. As an illustrative example, consider Fig. 1 which depicts the WE space with two dimensions: *Latent feature 1* and *Latent feature 2*. In this figure, each *dot* represents a word in the vocabulary and the *stars* represent the words in the document $D1$. The *square* represents the *mean* of the words in $D1$. Since the *mean* also lies in the same WE space, the document $D1$ is represented as another word in the WE space.

There is an opportunity to improve the document representation by representing the documents in a higher-dimensional feature space with interpretable word features. Consider a feature map $\phi$ : WS $\mapsto$ DS, where WS is the word embedding space and DS is the document space, and $\phi$ maps the documents which are *packed* in WS as overlapping word sets into DS as position vectors. In DS, each dimension corresponds to a feature word and a given document is represented with the corresponding distances from the set of potential feature words selected from WS. As an illustrative example, consider Fig. 2, which depicts DS with two dimensions: *Feature word 1* and *Feature word 2*. In this figure, each dot represents a document. For document $D1$ (which is a word set in WS as shown in Fig. 1), the distance from *Feature word 1* to $D1$ is $x1$, which becomes one component, and the distance from *Feature word 2* to $D1$ is $x2$, which becomes another component. So, under the proposed DIFW model,

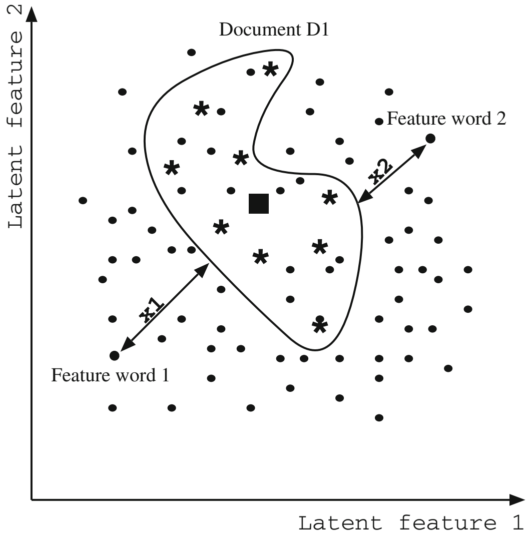

Fig. 1 Word embedding space (WS)

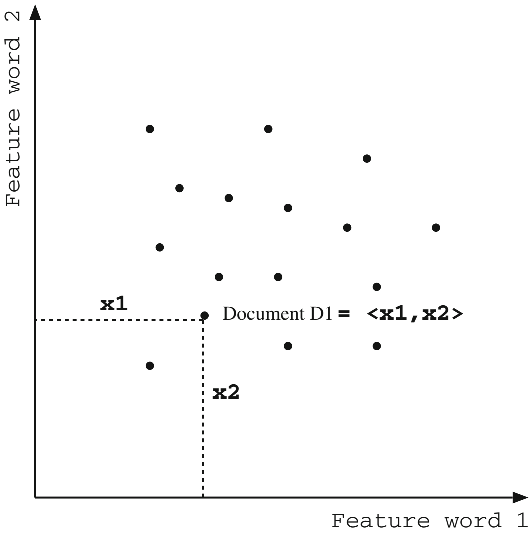

Fig. 2 Document space (DS)

like $D1$, a document is represented as a position vector in DS.

The document representation model under the proposed DIFW framework is as follows. Consider a set of words $\{w_1, w_2, \ldots, w_n\}$ of size $n$, sampled from the vocabulary of given set of documents. A given document $D$ is modeled as a n-ary vector $< d_1, d_2, \ldots, d_n >$ s.t. $d_i \in N_i$. Here, $N_i$ is a domain of distance values corresponding to the feature word $w_i$, and $d_i$ is a distance value in domain $N_i$, which represents the distance between $w_i$ and $D$.

In WE space, a word selected as a feature word acts as a representative for the concept that is conveyed by its neighboring words and the distance represents the degree of semantic dissimilarity. So under the DIFW framework, the document representations expressed in terms of distances from multiple feature words possess high representational power along with the advantage of feature interpretability.

Given the dataset of documents and pre-trained WEs, the two main steps in the DIFW model to represent the given document are as follows: (1) selection of feature words from the vocabulary, and (2) measuring the distance between feature words and the given document. We present the corresponding approaches in the following subsections.

## 3.2 Approaches to select feature words

We present two approaches for the selection of feature words from the vocabulary formed by the given dataset. First, we explain the approach based on the words frequency distribution. Next, we present the approach based on the words spatial distribution.

### 3.2.1 Words frequency distribution-based approach

In natural language, words occur according to the Zipf's law [41]. As a result, frequency distribution of *words* is a long tail distribution [2]. As an example, consider a sample frequency distribution of *words* from 20*Newsgroups* dataset in Fig. 3. High-frequency words are positioned in the *Head* part of the curve, and the rare words are positioned in the *Long tail* of the curve. A rudimentary strategy to select feature words from the vocabulary is a random selection. A word drawn independently from the vocabulary at random is more likely to come from the long tail part of the distribution. So, most of the feature words will be *rare words*. The *rare words* possess greater specificity so collectively as features they can cover very few documents [38]. On the other end, *very high frequent words* posses low discriminative power and are the candidates for stop-words [35]. The words which are moderately frequent have both high discriminative power and high coverage of the documents. So, in this approach, the feature words are selected by choosing *moderately frequent words* from the vocabulary of the dataset.

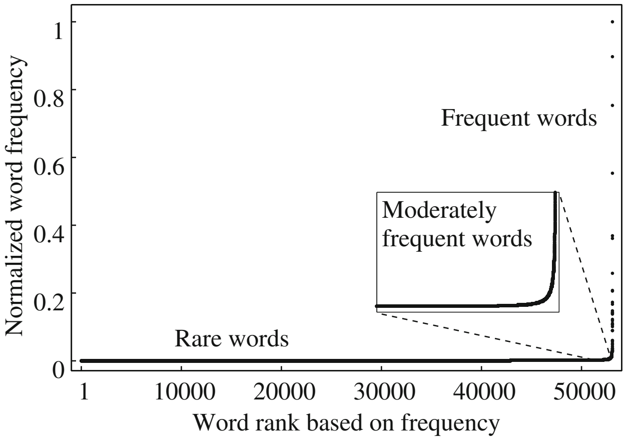

**Fig. 3** Words frequency distribution for 20*Newsgroups* dataset

### 3.2.2 Words spatial distribution-based approach

The WE models encode the relatedness of words as the spatial distances among them in WE space. So, if we consider a sample of domain-related words (DRWs) from different domains in WE space, the words from the same domain will be positioned closer to each other, but the words belonging to different domains will be positioned farther to each other.

In this section, along with the domain-related words (DRWs), we analyze the positioning of generic words (GWs) in the WE space. Here, a generic word is a word which is particularly not related to any domain but commonly occurs in all the domains.

To understand the positioning of words based on their domain-relatedness, consider $n$ DRWs $\overline{x}_1, \overline{x}_2, \overline{x}_3, ..., \overline{x}_n$ and an ideal GW $\overline{x}_{n+1}$ which co-occurs with all the DRWs uniformly. We can operationalize the distributional hypothesis over these words by defining the most general mean squared error cost function $E$ as shown in Eq. 1.

$$E = \frac{2}{n(n+1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n+1} (-1)^r \|\overline{x}_i - \overline{x}_j\|^2 \qquad (1)$$

The cost function $E$ is defined over the pairwise distances of all the words from $\overline{x}_1$ to $\overline{x}_{n+1}$. In $E$, $\|\overline{x}_i - \overline{x}_j\|$ is the euclidean distance between two words $\overline{x}_i$ and $\overline{x}_j$. Here, for related words $r = 0$, and for unrelated words $r = 1$. Thus, by minimizing $E$, we can minimize the distance between the related words and maximize the distance between unrelated words.

Let's say we use gradient descent optimization algorithm to minimize $E$. In gradient descent algorithm, all the word vectors (parameters) will be initialized at random and then iteratively updated by the gradients which minimize $E$. The gradient of $E$ at the generic word $\overline{x}_{n+1}$ can be calculated as follows.

$$\begin{aligned}\frac{\partial E}{\partial \overline{x}_{n+1}} &= \frac{4}{n(n+1)}\sum_{i=1}^{n}(\overline{x}_i - \overline{x}_{n+1})\\ &= \frac{4}{n(n+1)}\Big(\sum_{i=1}^{n}\overline{x}_i - \sum_{i=1}^{n}\overline{x}_{n+1}\Big)\\ &= \frac{4}{n+1}\Big(\frac{1}{n}\sum_{i=1}^{n}\overline{x}_i - \overline{x}_{n+1}\Big)\\ &= k(\mu - \overline{x}_{n+1}) \qquad (2)\end{aligned}$$

Here, $k$ is a constant and $\mu$ is the mean of all DRWs. From Eq. 2, we can say that to minimize $E$, the generic word vector $\overline{x}_{n+1}$ should be updated toward the mean of the DRWs. Similarly, if we consider an ideal DRW which is unrelated to most of the words in the vocabulary, its word vector will be updated away from the mean of unrelated words.

Alternatively, in $E$, if we consider mean absolute error instead of mean square error, the GW will be updated toward the geometric median [12] of DRWs. Here, note that both mean and median are the centrality measures. From this, we can say that to minimize $E$, GWs will be positioned near to the centrality measure, which we call as the *center* and DRWs will be positioned away from the center. For a given word, its distance from the center quantifies its degree of domain specificity.

Since the cost function $E$ that we considered is very generic, we can expect similar phenomena for the other WE models such as *Word2vec* [28] and GloVe [33] which also operationalize the distributional hypothesis. For simplicity, for the rest of the paper, we consider the mean of all the words in the vocabulary as the center and the distance between a word and the center as the word's *radius*.

Based on the radius, we can divide the words into three groups: closest words, distant words, extreme distant words. Closest words are GWs, and they possess low discriminative power. Distant words are DRWs and have relatively better discriminative power. Extreme distant words are also DRWs, but these words are loosely related to all the other words. So under this approach, the feature words are selected by choosing *DRWs excluding extreme distant words* from the vocabulary of the dataset.

As an example, consider the sample spatial distribution of words for 20*Newsgroups* dataset in Fig. 4. Here, $X$-axis represents the word rank based on the radius, and $Y$-axis represents the normalized radius of the word. Based on the contour patterns of the radius curve, it is divided into three groups. Words ranked below $a$ are closest words, words ranked between $a$ and $b$ are distant words in DRWs, and words ranked above $b$ are extreme distant words in DRWs.

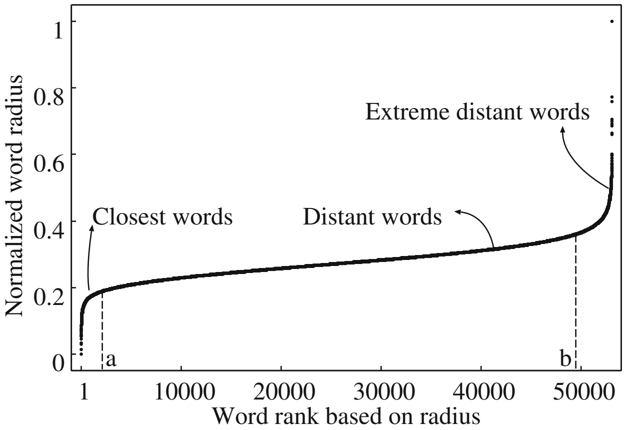

**Fig. 4** Words spatial distribution for 20*Newsgroups* dataset

### 3.3 Word-document distance function

In this section, we present an approach to compute the spatial distance between a given word and a document.

The distance between any two entities in a vector space quantifies the similarity (or dissimilarity) between them. A distance measure can be defined in multiple ways, but the most important property of a distance function is to have large discriminative power [11]. In the literature regarding agglomerative hierarchical cluster analysis [18] and object matching [11], there have been many distance measures defined to find the distance between any two point sets. By considering the word as point and document as a point set, there is an opportunity to define new distance measures in the WE space. We first discuss the distance measures in cluster analysis and object matching and explain the proposed distance measures.

In agglomerative hierarchical clustering, at each step, two most similar clusters are merged into a single cluster. Here, a cluster is a set of data point vectors. The similarity (proximity) between the two clusters can be computed by using one of the following measures: single-linkage, complete-linkage, and average-linkage. In single-linkage clustering, the distance between two clusters is defined as the distance between the two closest pair of points where each point belongs to a different cluster [18,36]. In complete-linkage clustering, the distance between two clusters is defined as the distance between the two farthest pair of points where each point belongs to a different cluster [18,23]. In average-linkage clustering, the distance between the clusters is defined as the average of all distances between the members of a cluster to all the members of other cluster [37].

It can be observed that single-linkage clustering merges two clusters into one if a member of a cluster is highly similar to at least one member of other cluster [4]. As the clusters merging criterion in single-linkage clustering is local [27], it favors long chain-like clusters. On the other hand, complete-linkage clustering merges two clusters into one if all the members of a cluster are highly similar to all the members of other cluster [4]. As the clusters merging criterion in complete-linkage clustering is non-local [27], it favors spherical and compact clusters. The average-linkage clustering strikes the compromise between the chaining tendency of single-linkage clustering and compacting tendency of complete-linkage clustering.

Similar to the cluster analysis, in the case of object (image) matching also an object is identified as a set of edge points. The purpose of object matching is to find the similarity between the two binary object images. The similarity between two object shapes, i.e., two edge point sets, is computed by finding the distance between them. Hausdorff distance is a popular measure to find the spatial distance between two point sets. It is defined as the maximum of the minimum distance between two sets of objects [15,32]. This measure is sensitive to noise so in [11], authors proposed modified distance definitions based on the Hausdorff distance by considering the mean and median of the minimum distances between two sets of objects.

The adaptation of preceding distance measures for the word-document case is as follows. By considering word and document as a single-element set and multi-element set, respectively, the distance measures discussed in the case of cluster analysis and object matching can be adapted to word-document case majorly in four ways. Let's say $w$ and $D$ are the given word and document, respectively. Now, in $D$, consider words $w_{\mathrm{c}}$, $w_{\mathrm{m}}$, and $w_{\mathrm{f}}$ such that they are the closet word, middle(median) word, and farthest word, respectively, from $w$. Given $D$ and $w$, words $w_{\mathrm{c}}$, $w_{\mathrm{m}}$, and $w_{\mathrm{f}}$ can be easily identified by computing the distances of all words in $D$ from $w$ and sorting the words in ascending order based on the distance values.

Now, the distance between $w$ and $D$ can be defined by four distance measures: $\overline{ww}_{\mathrm{c}}$, $\overline{ww}_{\mathrm{m}}$, $\overline{ww}_{\mathrm{f}}$, and $d_{\mu}$ which are adapted from single-linkage, modified Hausdorff distance, complete-linkage, and average-linkage, respectively. The measure $\overline{ww}_{\mathrm{c}}$ defines the distance between the word $w$ and document $D$ as the distance between distance between word $w$ and the closest word $w_{\mathrm{c}}$ in $D$. The measure $\overline{ww}_{\mathrm{m}}$ defines the distance between the word $w$ and document $D$ as the distance between distance between word $w$ and the middle word $w_{\mathrm{m}}$ in $D$. The measure $\overline{ww}_{\mathrm{f}}$ defines the distance between the word $w$ and document $D$ as the distance between distance between word $w$ and the farthest word $w_{\mathrm{f}}$ in $D$. The measure $d_{\mu}$ defines the distance between the word $w$ and document $D$ as the mean of all the distances from $w$ to every word in $D$.

$$\overline{ww}_{\mathrm{c}} = \mathrm{dist}(\overline{w}, \overline{w}_{\mathrm{c}}) \tag{3}$$

$$\overline{ww}_{\mathrm{m}} = \mathrm{dist}(\overline{w}, \overline{w}_{\mathrm{m}}) \tag{4}$$

$$\overline{ww}_{\mathrm{f}} = \mathrm{dist}(\overline{w}, \overline{w}_{\mathrm{f}}) \tag{5}$$

$$d_{\mu} = \frac{1}{l} \sum_{i=1}^{l} \mathrm{dist}(\overline{w}, \overline{w}_i) \tag{6}$$

Here, $l$ is the size of the document and $\mathrm{dist}(\overline{w}, \overline{w}_i)$ is a spatial distance measure (such as $L_1$-norm distance or $L_2$-norm distance) to find the distance between two word vectors $\overline{w}$ and $\overline{w}_i$ in WE space.

Even though these distance definitions are simple and parameter-free, they suffer from the following limitations. In the cases of $\overline{ww}_{\mathrm{c}}$, $\overline{ww}_{\mathrm{m}}$, and $\overline{ww}_{\mathrm{f}}$ (Eqs. 3, 4, 5), much information is lost due to selecting a single word in D and finding the distance between that word and $w$ and also these measures are sensitive to outliers and noisy words. In the case of $d_{\mu}$ (Eq. 6), all distances are averaged together by ignoring the fact that the words with different semantic meanings are positioned at different distances from $w$. Overall, the preceding distance measures have low discriminative power as they carry very low information content.

We propose improved distance measures to overcome the limitations of word-level distances (Eqs. 3, 4, 5, 6). It can be observed that in WE space, words belonging to the same neighborhood represent the same concept or topic [20,21]. A topic, as compared to word, can carry higher information content and it is more immune to outliers. By extending the word-level distances, we propose new distance definitions by considering the notion of topic.

Consider $T_{\mathrm{c}}$, $T_{\mathrm{m}}$, and $T_{\mathrm{f}}$, as the closest, middle, and farthest topics in the given document $D$, respectively. Notably, computing words belong to $T_{\mathrm{c}}/T_{\mathrm{m}}/T_{\mathrm{f}}$ is simple, once we know the corresponding $w_{\mathrm{c}}/w_{\mathrm{m}}/w_{\mathrm{f}}$. The topics $T_{\mathrm{c}}$, $T_{\mathrm{m}}$, and $T_{\mathrm{f}}$ are the collection of words neighboring $w_{\mathrm{c}}$, $w_{\mathrm{m}}$, and $w_{\mathrm{f}}$, respectively. The distance between $w$ and $T_{\mathrm{c}}/T_{\mathrm{m}}/T_{\mathrm{f}}$ can be calculated by simply averaging the distances from $w$ to the words in $T_{\mathrm{c}}/T_{\mathrm{m}}/T_{\mathrm{f}}$.

The proposed distance function measures the distance between word $w$ and document $D$ as the overall deviation of $D$ from the $w$. The deviation of document $D$ from the word $w$ can be computed as the root mean squared distances from $w$ to $T_{\mathrm{c}}$, $T_{\mathrm{m}}$, and $T_{\mathrm{f}}$ in $D$. The formal definition of the proposed distance function to find the distance between a feature word and document is as follows. Consider a feature word $\overline{w}$ and a document $D$ which consists of $l$ words. Let $D = \langle \overline{w}_1, \overline{w}_2, \overline{w}_3, \ldots, \overline{w}_l \rangle$ represent the ordered list of $l$ words such that the words in $D$ are arranged in the ascending order of their distance from $\overline{w}$, i.e., $\mathrm{dist}(\overline{w}, \overline{w}_i) \leq \mathrm{dist}(\overline{w}, \overline{w}_j), \forall\ i < j,\ 1 \leq i, j \leq l$. Also, let $T_{\mathrm{c}}$, $T_{\mathrm{m}}$, and $T_{\mathrm{f}}$ be the closet topic, median topic and the farthest topic, respectively. The sizes of these topics are $|T_{\mathrm{c}}| = \frac{\alpha}{100} \times l$, $|T_{\mathrm{m}}| = \frac{\beta}{100} \times l$, and $|T_{\mathrm{f}}| = \frac{\gamma}{100} \times l$, where $\alpha$, $\beta$, and $\gamma$ are positive real values.

Let DC (distance between $\overline{w}$ and the closest topic $T_{\mathrm{c}}$), DM (distance between $\overline{w}$ and the middle topic $T_{\mathrm{m}}$), and DF (dis-

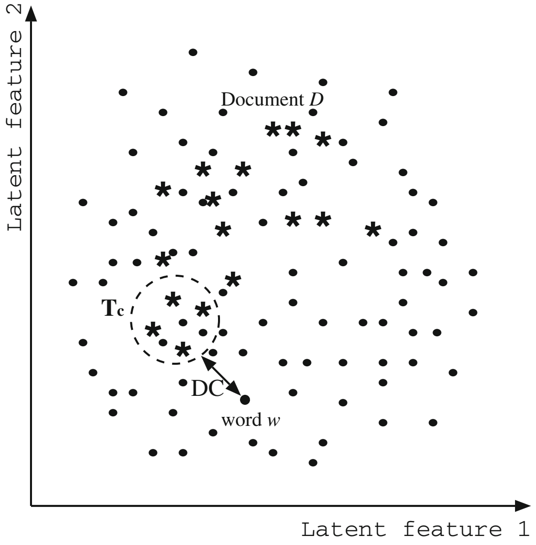

**Fig. 5** DC distance measure

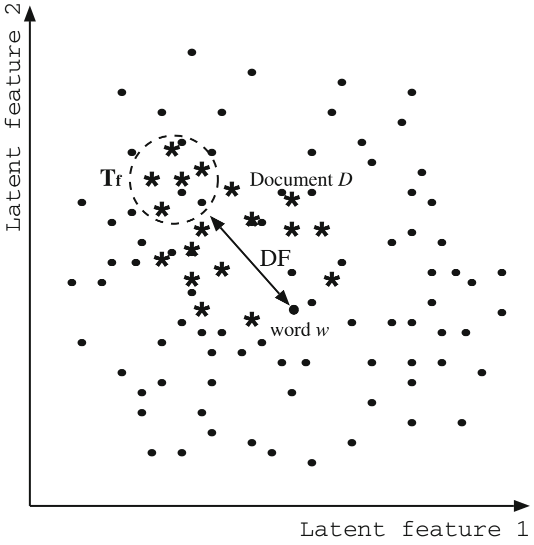

**Fig. 6** DF distance measure

tance between $\overline{w}$ and the farthest topic $T_\mathrm{f}$) represent distance measures to measure the distance from $\overline{w}$ and $D$. Also, let DA (distance based on all topics) distance measure represents the proposed distance measure. The formulas for computing DC, DM, DF and DA are given in Eqs. 7, 8, 9 and 10, respectively. Here, DA is the proposed word-document distance function. Given $\overline{w}$ and D, we need the values of $\alpha$, $\beta$ and $\gamma$ to determining $|T_\mathrm{c}|$, $|T_\mathrm{m}|$ and $|T_\mathrm{f}|$, which are the dataset dependent.

$$\mathrm{DC} = \frac{1}{|T_\mathrm{c}|}\sum_{i=1}^{|T_\mathrm{c}|}\mathrm{dist}(\overline{w}, \overline{w}_i), \quad \text{where } 1 \le |T_\mathrm{c}| \le l \tag{7}$$

$$\mathrm{DM} = \frac{1}{|T_\mathrm{m}|}\sum_{i=\frac{l}{2}-\frac{|T_\mathrm{m}|}{2}}^{\frac{l}{2}+\frac{|T_\mathrm{m}|}{2}}\mathrm{dist}(\overline{w}, \overline{w}_i), \quad \text{where } 1 \le |T_\mathrm{m}| \le l \tag{8}$$

$$\mathrm{DF} = \frac{1}{|T_\mathrm{f}|}\sum_{i=l-|T_\mathrm{f}|+1}^{l}\mathrm{dist}(\overline{w}, \overline{w}_i), \quad \text{where } 1 \le |T_\mathrm{f}| \le l \tag{9}$$

$$\mathrm{DA} = \mathrm{RMS}\big(\mathrm{DC}, \mathrm{DM}, \mathrm{DF}\big) = \sqrt{\frac{\mathrm{DC}^2 + \mathrm{DM}^2 + \mathrm{DF}^2}{3}} \tag{10}$$

The detailed definitions and characteristics of DC, DM, DF, and DA are as follows.

DC defines the distance between the word and document as the distance between the word and the closest topic in the document. DC is the generalized version of the distance measure in Eq. 3. DC is also inspired by single-linkage clustering, so it inherits the properties of single-linkage clustering. Similar to single-linkage clustering, the DC measure's criterion is local (see Fig. 5). DC measure supposes that the word and document are related if at least one topic in the document is related to the word and it doesn't pay attention to the rest of the document (see Fig. 5). This property of DC measure makes it suitable for the multi-label document representation where multiple topics are discussed in the single document.

DF defines the distance between the word and document as the distance between the word and the farthest topic in the document. DF is the generalized version of the distance measure in Eq. 5. DF is also inspired by complete-linkage clustering, so it inherits the properties of complete-linkage clustering. Similar to complete-linkage clustering, the DF measure's criterion is non-local (see Fig. 6). DC measure supposes that the word and document are related if and only if all the topics in the document are related to the word (see Fig. 6). This property of the DF measure makes it suitable for the single-label document representation where the whole document is about a single topic.

DM defines the distance between the word and document as the distance between the word and the middle (median) topic in the document. DM is the generalized version of the distance measure in Eq. 4. The DM measure tries to strike the compromise between DC and DF measures. DM measure supposes that the word and document are related if the majority (at least half) of the topics are related to the word (see Fig. 7).

The proposed DA measure defines the distance between the word and document as the root mean square (RMS) value of the distances DC, DM, and DF. It serves as an aggregator of magnitudes of DC, DM, and DF measures. Unlike DC, DM, and DF, it doesn't calculate the distance based on a single aspect of the document; instead, it calculates the overall spread (deviation) of the document from the word in terms of the topical distances (see Fig. 8). Therefore, one can notice

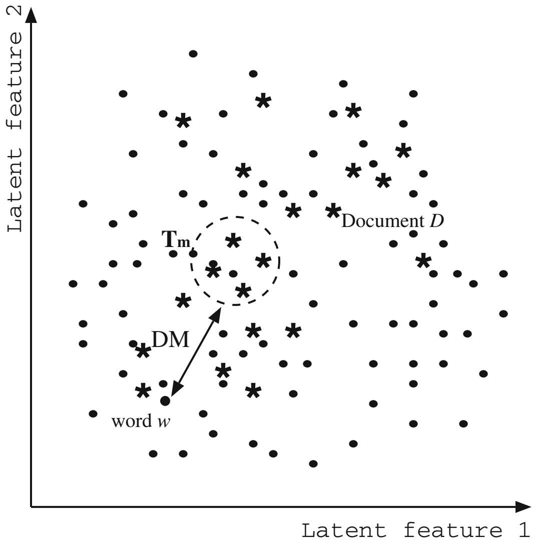

**Fig. 7** DM distance measure

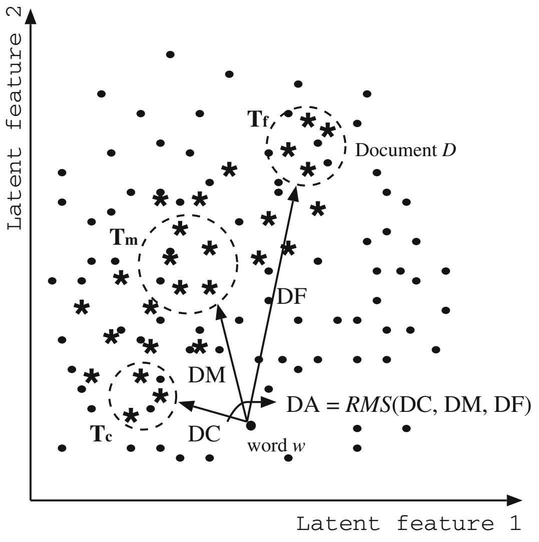

**Fig. 8** DA distance measure

the resemblance between the DA measure (Eq. 10) and the standard deviation formula.

Overall, the DA measure is simple and carries higher information content as compared to all the previously discussed distance measures. Furthermore, in contrast, to mean distance (Eq. 6), it captures the semantic meanings of the individual words as only semantically related word's distances are averaged together (topics), so it can be interpreted as a form of weighted averaging with word weights assigned according to their topic sizes [17]. For these reasons, the proposed distance measure has higher discriminative power over other distance measures.

## 4 Experimental results

We have conducted experiments on 4 different text classification datasets. The $20Newsgroups$[1] dataset is a collection of news articles classified into 20 categories. We removed metadata such as headers, signatures, and quotations from the documents which act as direct clues to the classes to make it more practical for text categorization. The $Reuters$[2] is a multi-class multi-label text classification dataset. For both $20Newsgroups$ and $Reuters$, the training set and test set split are predefined. The $BBC$[3] dataset contains news stories from 2004 to 2005. The $AGNews$ is originally created by Zhang et al. [40] using a large collection of titles and description fields of news articles. For computational efficiency, we random sampled $AGNews$ dataset. We created the training set and test set for the $BBC$ and $AGNews$ datasets with the split ratio of 60:40.

In the experiments, for WEs, we used publicly available GloVe vectors.[4] These WEs are of 300 dimensions and trained on a collection of Wikipedia articles. While implementing the models which require pre-trained WEs, the words whose pre-trained WEs are not available are dropped from the vocabulary. For the models which are independent of pre-trained WEs, we considered the whole vocabulary.

Table 1 contains the details of the datasets. It contains the details such as the number of documents ($N$), actual vocabulary size ($|V|$), vocabulary size after dropping words whose embeddings are not available ($|V'|$), average document length $D$, average document length after dropping words whose WEs are not available ($D'$), and the number of classes.

For all experiments, we have removed the stop-words from the documents before their feature vectors generation. We used linear SVM classifier to perform the classification task. We used computationally inexpensive $L_1$-norm distance measure to find the distance between two words to compute the distance for the distance measures DC, DM, DF, and DA. The default values of $\alpha$, $\beta$, and $\gamma$ in DA measure are 3%, 70%, and 15%, respectively.

We have conducted the following experiments.

– we have evaluated the performance of the proposed DIFW model using the feature words selection approach based on the words frequency distribution. We call this model as DIFW-fd model for the rest of the paper.
– we have evaluated the performance of the proposed DIFW model using feature words selection approach

[1] http://qwone.com/~jason/20Newsgroups/.

[2] https://archive.ics.uci.edu/ml/machine-learning-databases/reuters21578-mld/.

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

[4] https://nlp.stanford.edu/projects/glove/.

**Table 1** Datasets details

| Dataset | $N$ | $\lvert V\rvert$ | $\lvert V'\rvert$ | $\lvert D\rvert$ | $\lvert D'\rvert$ | Classes |
|---|---|---|---|---|---|---|
| 20Newsgroups | 18,846 | 148,060 | 53,089 | 170 | 158 | 20 |
| Reuters | 10,788 | 42,289 | 27,555 | 102 | 99 | 90 |
| AG News | 19,000 | 31,382 | 26,514 | 32 | 31 | 4 |
| BBC | 2225 | 17,746 | 10,646 | 100 | 78 | 5 |

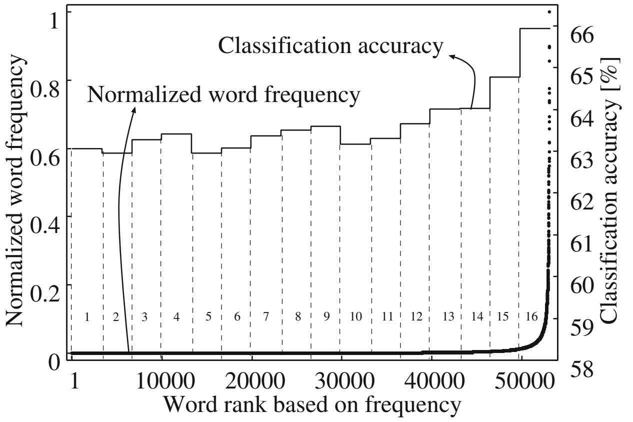

**Fig. 9** Performance of DIFW-fd on 20*Newsgroups* dataset

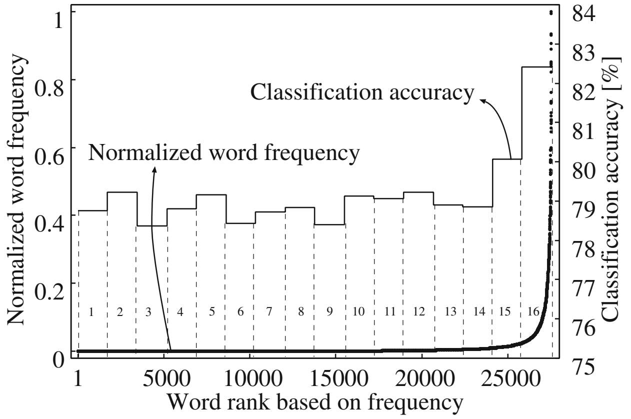

**Fig. 10** Performance of DIFW-fd on *Reuters* dataset

based on words spatial distribution. We refer this model as DIFW-sd model for the rest of the paper.
- Based on experimental logs, we provided the qualitative analysis of spatial distribution of words.
- The performance analysis of distance measures DC, DM, DF, and the proposed DA measure by varying the values of hyper-parameters $\alpha$, $\beta$, and $\gamma$.
- Performance comparison of the proposed DIFW-fd and DIFW-sd models against 8 baseline methods.
- Performance analysis of hyper-parameters: the number of feature words selected from the vocabulary ($n$), closest topic size ($\alpha$), median topic size ($\beta$), and farthest topic size ($\gamma$).

### 4.1 Performance analysis of DIFW-fd

We demonstrated the performance analysis of DIFW-fd on 20*Newsgroups* and *Reuters* datasets. Figure 9 shows the quantitative analysis of DIFW-fd on 20*Newsgroups* dataset. In this experiment, the distribution curve is formed by raking the words in the vocabulary in ascending order of their *frequency*. The distribution curve of DIFW-fd is divided into 16 equal-sized bins such that each bin contains the words approximately equal to 6.2% of the vocabulary size($|V'|$). The size of the bins is chosen such that the trends in the distribution curves are well separated by the bins. For each bin, using the feature words coupled with DA measure, the vector representations of documents in both training set and test set are generated. Linear SVM classifier is trained on the training set vector representations, and classification task is performed on the test set. The classification accuracies over the test set for all the bins are shown in Fig. 9. The figure contains two curves: the normalized word frequency curve corresponds to primary $Y$-axis, and the classification accuracy step curve corresponds to secondary $Y$-axis. For the first curve, $X$-axis represents the rank of the words based on frequency, and for the second curve, $X$-axis represents the bin number. The bin numbers are indicated as 1–16 in the graph.

From Fig. 9, it can be observed that the overall performance is increasing with the bin number. Bins 1–13 contain very rare words with a comparable frequency. As a result, the accuracies for these bins are relatively low and follow an arbitrary trend. From 14th bin to 16th bin, the frequency of words increases. Since the stop-words are removed, the last bin (16th) contains moderately frequent words in the dataset which have high discriminating power and high documents coverage. So the accuracies for these bins are consistently increasing and the last bin exhibits the highest performance.

The results for *Reuters* dataset are shown in Fig. 10. Notably, a similar trend has been exhibited for *Reuters* dataset. We have also conducted experiments for other datasets and obtained similar results. From this experiment, for the given dataset, we can conclude that the proposed approach exhibits maximum performance with moderately frequent words as feature words.

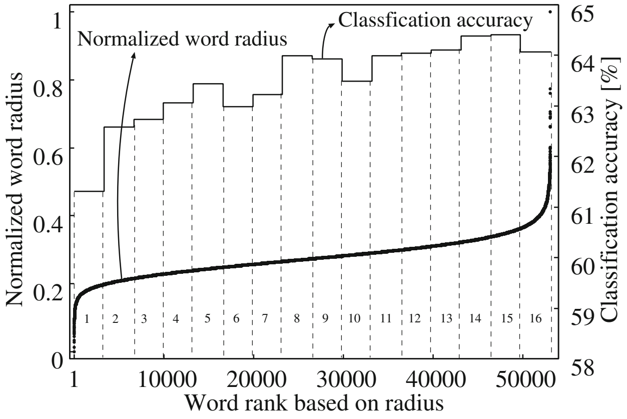

**Fig. 11** Performance of DIFW-sd on 20$Newsgroups$ dataset

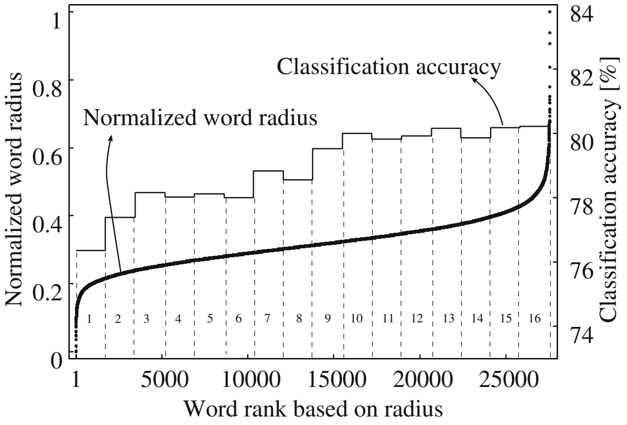

**Fig. 12** Performance of DIFW-sd on $Reuters$ dataset

### 4.2 Performance analysis of DIFW-sd

We demonstrated the performance analysis of DIFW-sd on 20$Newsgroups$ and $Reuters$ datasets. Figure 11 shows the qualitative analysis of DIFW-sd for 20$Newsgroups$ dataset. In this experiment, the distribution curve is formed by ranking the words in the vocabulary in ascending order of their $radius$. Similar to the preceding experiment, the distribution curve of DIFW-sd is divided into 16 equal-sized bins such that each bin contains words around 6.2% of the vocabulary size($|V'|$). For each bin, using the feature words coupled with DA measure, the vector representations of documents in both training set and test set are generated.

The classification accuracies over the test set for all the bins corresponding to 20$Newsgroups$ dataset are shown in Fig. 11. The figure contains two curves: the normalized word radius curve corresponds to primary $Y$-axis, and the classification accuracy step curve corresponds to secondary $Y$-axis. For the first curve, $X$-axis represents the rank of the words based on radius, and for the second curve, $X$-axis represents the bin number.

From the figure, it can be observed that the overall performance is increasing with the bin number. The first bin contains the GWs which have the least discriminative power, so they have the lowest classification accuracy. Bins 2–14 contain DRWs with slightly increasing radius. It can be observed that the accuracies for these bins are overall exhibiting an increasing trend. The last bin, which is bin 16, also contains the DRWs. But, these are extreme distant words and loosely related to the other words. Bin 15 contains the words which are ranked before the extreme distant words and has the highest performance.

The results for $Reuters$ dataset are shown in the Fig. 12. It can be noted that the overall performance is increasing with the radius. The lowest performance is exhibited by GWs in the first bin. However, notably in this case, unlike in 20$Newsgroups$ dataset case, the extreme distant words in the 16th bin perform slightly better than the DRWs in the 15th bin. Normally, the extreme distant words contain noisy words, which occur out of context in the dataset. As a result, the performance of the bin containing the extreme distant words could be arbitrary. From this, we can say that it is safe to choose the words in the 15th bin which are DRWs and are ranked before the extreme distant words as feature words. We have also conducted experiments for other datasets and obtained similar results.

From this experiment, for a given dataset, we can conclude that the proposed approach exhibits maximum performance with farthest DRWs, which are ranked before extreme distant words as feature words.

### 4.3 Qualitative analysis of spatial distribution of words

For the qualitative analysis of spatial distribution of words, in Table 2, we listed 25 closest words to the center and 25 farthest words from the center along with their frequencies from 20 $Newsgroup$ dataset. From the lists, we can clearly observe that the words which are closest to the center are GWs and words farthest from the center are DRWs. For example, the word *lastly* is a generic word and not related to any particular domain, whereas the word *republish* is not a generic word and it is related to *literature* domain. From the lists, we can also observe that the frequencies of GWs have a wide range and they do not come under stop-words as well as rare words. So we can’t filter them from the vocabulary by using the traditional frequency-based trimming. However, using radius as a measure we can easily identify the GWs.

### 4.4 Performance analysis of distance measures

In this experiment, we analyze the performance of distance measures DC, DM, DF, and DA. The experimental analysis is presented for 20$Newsgroups$ and $Reuters$ datasets using

**Table 2** Qualitative analysis of spatial distribution of words

| Words close to the center |
|---|
| lastly (8), likewise (30), interestingly (18), moreover (38), incidentally (24), furthermore (81), ironically (12), conversely (4), instance (143), indeed (270), aforementioned (9), presumably (67), importantly (19), wherein (10), evidenced (8), implying (16), consequently (11), coincidentally (3), unfortunately (236), whereupon (3), evidently (15), i.e (208), meantime (20), additionally (24), characterizing (2), paradoxically (4) |
| Words far away from center |
| republish (1), nytimes (4), daybook (1), unalienable (3), wildcards (1), distillates (1), incrimination (1), linescores (1), near-earth (4), teaspoon (2), bushel (1), advisories (1), resending (2), amortisation (1), polynomial (5), multi-engine (1), affine (2), polytopes (2), projective (1), prohibitive (2), excerpted (3), autosomal (2), tensor (3), undocked (1), genus (1), backhand (4), megabits (1) |

both feature words selection approaches. For each dataset, we select feature words based on one of the approaches. Next, we generate the vector representations for the documents with the distance measures DC, DM, DF, and DA. Further, we use these document representations to perform the classification task and compare the distance measures by classification accuracies. The number of feature words selected ($n$) for each dataset may vary, but it kept constant for all the distance measures comparison experiments conducted on the same dataset. The distance measures DC, DM, and DF have one hyper-parameter in their definitions, which are $\alpha$, $\beta$, and $\gamma$, respectively. To find the maximum performance of these distance measures, the corresponding values of $\alpha$, $\beta$, and $\gamma$ are varied. In the proposed distance measure DA also, we analyzed the performance by varying the three hyper-parameters $\alpha$, $\beta$, and $\gamma$. However, for visualization simplicity, we reported results of DA at the default values of $\alpha$, $\beta$, and $\gamma$.

Figure 13 shows the performance of distance measures DC, DM, and DF with feature words selected based on word frequency distribution approach on 20*Newsgroups* dataset. We selected 4500 (about 8% of $|V'|$) moderately frequent words as feature words. To analyze the performance of DC, DM, and DF, for each document, the sizes of closest topic $|T_\mathrm{c}|$, median topic $|T_\mathrm{m}|$ and farthest topic $|T_\mathrm{f}|$ are varied by varying the corresponding values of $\alpha, \beta, \gamma$ from 1% to 100% of the document size.

It can be observed that at 1% of $\alpha/\beta/\gamma$, the performance of DC/DM/DF is 52%, 54%, and 57%. The performance of DC/DM/DF exhibits different trends with increasing $\alpha/\beta/\gamma$. As $\alpha$ increases, the performance of DC increases to a peak when $\alpha = 7\%$, and then decreases and settles at about 61%.

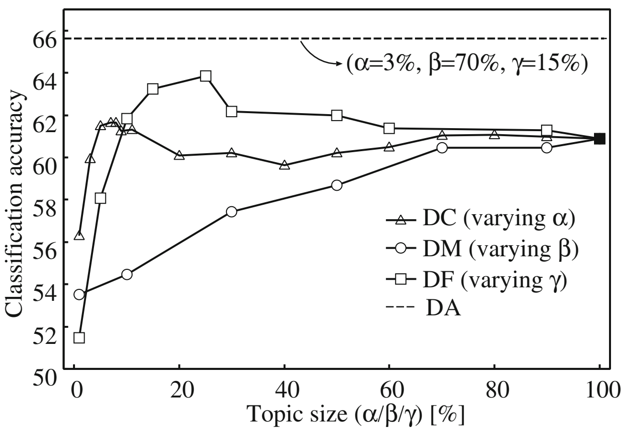

**Fig. 13** Performance of distance measures with feature words selected by word frequency distribution approach on 20*Newsgroups* dataset

This indicates that the highest performance of DC could be achieved at the smaller values of $\alpha$ up to 10%. As $\beta$ increases, the performance of DM rapidly increases until $\beta = 70\%$ and settles at about 61%. This indicates that the highest performance of DM could be achieved with the values of $\beta$ above 70%. As $\gamma$ increases, the performance DF increases to peak when $\gamma = 25\%$ and then decreases and settles at about 61%. This indicates that the highest performance of DF could be improved with the values of $\gamma$ around 25%. The performance of DC/DM/DF settles at 61% when $\alpha = \beta = \gamma = 100\%$ because all the words of the document are covered by each approach.

Figure 13 also shows the performance of the distance measure DA. To analyze the performance of DA for each document, the sizes of closest topic $|T_\mathrm{c}|$, median topic $|T_\mathrm{m}|$ and farthest topic $|T_\mathrm{f}|$ are fixed at $\alpha = 3\%$, $\beta = 70\%$, $\gamma = 15\%$ after conducting experiments at different values of $\alpha$, $\beta$, and $\gamma$. The results show that the DA improves the performance significantly over DC, DM, and DF. It can be noted that the performance obtained by DC, DM, and DF could not achieve the maximum performance as that of DA, even though the corresponding values of $\alpha$, $\beta$ and $\gamma$ are varied by the whole possible range.

Figure 14 shows the performance of distance measures with feature words selected by the words spatial distribution approach on 20*Newsgroups* dataset. We selected 4500 farthest DRWs excluding extreme distant words as feature words. Figures 15 and 16 show the performance of distance measures with feature words selected by word frequency distribution approach and word spatial distribution approach, respectively, on *Reuters* dataset. For *Reuters* dataset, we selected 2500 (about 9% of $|V'|$) feature words for both feature words selection approaches.

Overall, the results in Figs. 13, 14, 15, and 16 demonstrate that DC performs well for small values of $\alpha$ (below 10%), DM performs well for very high values of $\beta$ (70–100%), and

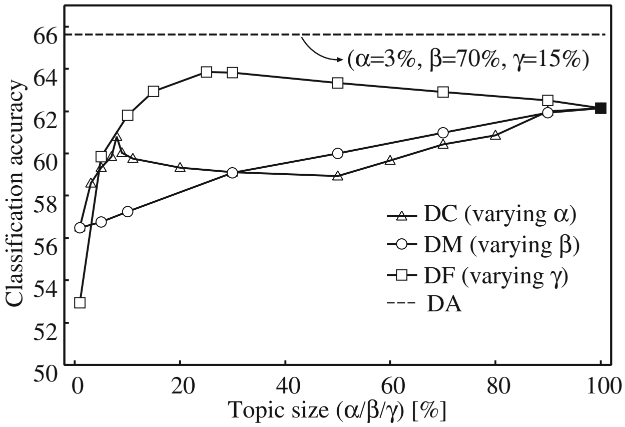

**Fig. 14** Performance of distance measures with feature words selected by word spatial distribution approach on 20*Newsgroups* dataset

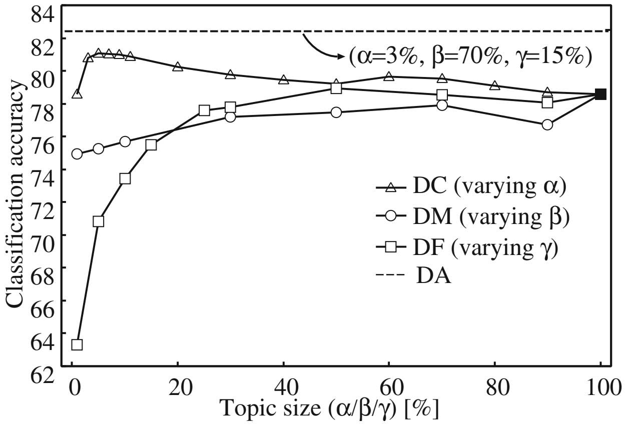

**Fig. 15** Performance of distance measures with feature words selected by word frequency distribution approach on *Reuters* dataset

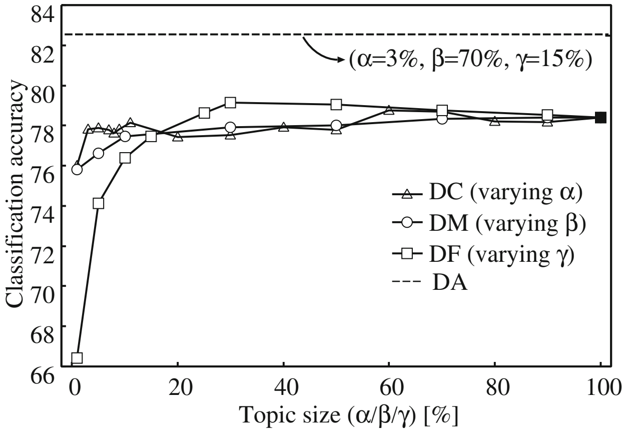

**Fig. 16** Performance of distance measures with feature words selected by word spatial distribution approach on *Reuters* dataset

DF performs well for medium values of $\gamma$ (25–50%). Usually, DF performs better and DM performs poorly among the three distance measures. The proposed DA measure fuses these three measures in its definition and exploits the individual powers of DC, DM, and DF. Hence, DA performs better than all of them irrespective of the feature words selection approach.

### 4.5 Performance comparison with baseline approaches

We have compared the proposed approach against 8 baseline methods on 4 different datasets. The baseline methods are vector averaging, Min–Max concatenation [10], SIF-embeddings [1], Doc2vec [25], Bag of words (BOW) [14], latent Dirichlet allocation (LDA) [5], Bag of concepts [20], Skip-thought vectors [22]. The datasets are 20*Newsgroups*, *Reuters*, *AGNews*, and *BBC*.

We used classification accuracy as the performance metric for all the datasets. The *Reuters* dataset is a multi-class multi-label dataset, so we used $F1$-score as another performance metric for this dataset. In all the experiments, threefold cross-validation is employed to tune the hyper-parameters.

In vector averaging, the document vector size is equal to the number of dimensions in word embeddings which is 300. In Min–Max concatenation, the document vector size is 600 where the first 300 dimensions are from the min vector and the rest of 300 dimensions are from the max vector. In SIF-embeddings, similar to vector averaging, the document vector size is 300. The weighting scheme in SIF-embeddings has a hyper-parameter $a$. The values of weighting parameter $a$ are chosen from $[10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$ for hyper-parameter tuning.

In Doc2vec,[5] we employed the distributed memory (dm) model to generate the document vectors. To tune the document vector size, its value is varied from 100 to 500 with step size 100. The number of negative samples drawn, window size, and the minimum count of words are 5, 8, and 5, respectively. The number of epochs hyper-parameter is tuned by choosing its values from [10, 50, 100, 200]. In Bag of words,[6] the number of dimensions is equal to the vocabulary size. In LDA,[7] the number of topics is hyper-parameter. To tune the number of topics, its values are varied from 100 to 600 with step size 100. In Bag of concepts, the number of concepts (or clusters) is a hyper-parameter and its values are varied from 1000 to 5000 with a step size of 500. For Skip-thought vectors, we used publicly available encoder model[8] which is pre-trained on large external book corpora. It takes the text documents as input and produces 4800-dimensional fixed-length vector for each sentence in a document which

[5] https://radimrehurek.com/gensim/models/doc2vec.html#gensim.models.doc2vec.Doc2Vec.

[6] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html.

[7] https://pypi.org/project/lda/.

[8] https://github.com/ryankiros/skip-thoughts.

**Table 3** Experimental results against baseline approaches

| | 20Newsgroups | Reuters accuracy | Reuters $F1$-Score | AG News | BBC |
|---|---|---|---|---|---|
| DIFW-fd | 65.61 | 82.49 | 47.35 | 90.03 | 93.22 |
| DIFW-sd | 64.21 | 81.06 | 45.52 | 89.26 | 93.89 |
| Vector averaging | 62.39 | 77.70 | 37.54 | 88.97 | 92.42 |
| Min–Max concatenation | 62.50 | 73.91 | 42.09 | 85.53 | 90.50 |
| SIF-embeddings | 63.02 | 77.77 | 37.37 | 89.17 | 92.65 |
| Doc2vec | 54.72 | 49.15 | 18.99 | 73.51 | 89.49 |
| BOW | 56.39 | 78.96 | 48.51 | 87.19 | 86.29 |
| LDA | 62.73 | 73.76 | 35.58 | 87.17 | 92.13 |
| Bag of concepts | 58.93 | 79.10 | 43.89 | 88.22 | 89.60 |
| Skip-thought | 53.87 | 72.04 | 34.32 | 81.96 | 92.62 |

are then averaged to produce document vector. In DIFW-fd and DIFW-sd, the number of feature words ($n$) is varied from 5 to 15% of vocabulary size and the values of $\alpha$, $\beta$, and $\gamma$ in DA measure fixed at their default values.

The comparative results with baseline methods are shown in Table 3. The results show that the proposed DIFW-fd and DIFW-sd approaches consistently improve the performance over other approaches on all the datasets.

The performance gain of the proposed model over the vector averaging is significant for $Reuters$ and $20Newsgroups$ datasets. The number of classes in both these datasets is relatively high and very closely related to each other. It signifies that the proposed document representation framework (document representation model + feature words selection approaches + distance measure) is able to capture multiple aspects of the document in an effective manner. As a result, the proposed framework is exhibiting improved performance. Vector averaging is comparatively insensitive to noisy words in the vocabulary than the Min–Max concatenation method. So vector averaging is performing slightly better than Min–Max concatenation. SIF-embeddings assigns weights to word embeddings based on their frequency, so it is performing consistently better than vector averaging. The rest of the baselines are performing comparably to each other. The Doc2vec model comes under the class of neural language models. Neural language models requires huge amounts of the data to perform effectively. The Doc2vec model is performing poorly as it attempts to co-learn both word embeddings and document embeddings using only data at hand [24].

The following observations can be made from DIFW-fd and DIFW-sd. In DIFW-fd, a feature word is selected based on its frequency, i.e., the property of word derived from the given dataset. In DIFW-sd, a feature word is selected based on its radius, i.e., the property of the word derived from the word embeddings spatial distribution, which is independent of the dataset. DIFW-sd performed better than all the baseline methods and able to perform as close as to DIFW-fd even though the methods of selecting feature words in DIF-sd are completely different from those in DIF-fd. So, the spatial distribution-based methodology provides an alternative avenue to improve the performance of document related tasks. The detailed investigation will be carried out as a part of future work.

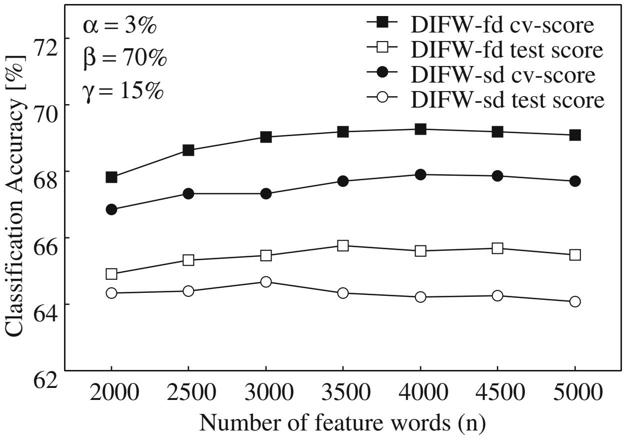

**Fig. 17** Analysis of hyper-parameter: $n$

## 4.6 Performance analysis of hyper-parameters

In our model, there are 4 hyper-parameters: the number of feature words ($n$) from the feature words selection approaches, the sizes of the closest topic ($\alpha$), median topic ($\beta$), and farthest topic($\gamma$) in terms of document length from the DA distance measure. We presented an empirical analysis of these parameters on $20Newsgroups$ dataset. the default values for $n$, $\alpha$, $\beta$, and $\gamma$ are 4500, 3%, 70%, and 15%, respectively. To analyze a parameter, we vary that parameter and fix the rest of the parameters at their default values.

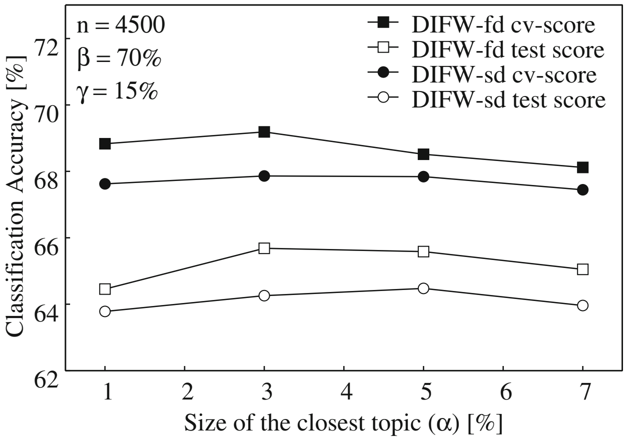

**Fig. 18** Analysis of hyper-parameter: $\alpha$

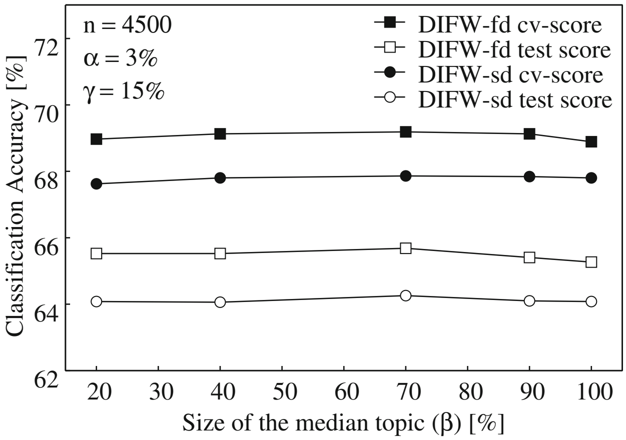

**Fig. 19** Analysis of hyper-parameter: $\beta$

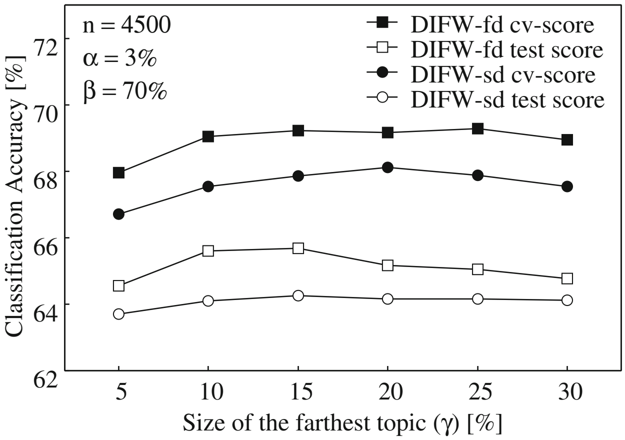

**Fig. 20** Analysis of hyper-parameter: $\gamma$

Figure 17 shows cross-validation score (cv-score) and test score for both DIFW-fd and DIFW-sd by varying number of feature words($n$) from 2000 to 5000. While increasing the $n$ value, at each step, the document representations are generated in computationally effective manner by simply appending the next 500 dimensions (corresponding to the next 500 most frequent feature words) to the previous vector representations. A similar procedure is followed while generating document vectors in DIFW-sd. With increasing $n$, the performance also increases and reaches the peak at a point and then slightly drops afterward. For both DIFW-fd and DIFW-sd, the cv-score increases till 4000 (about 7.5% of the vocabulary size) and drops afterward. The performance follows this trend because when $n$ value is small the number of feature words is not sufficient enough to represent the documents effectively, as the $n$ increases the expressive power of representations increases, after reaching the peak, as $n$ increases the non-discriminative words are added to the features and affects the performance negatively.

Figures 18, 19, and 20 show the cv-score and test score for both DIFW-fd and DIFW-sd by varying $\alpha$, $\beta$, and $\gamma$, respectively. From Fig. 18, we can observe that for the values of $\alpha$ the cv-score of DIFW-fd increases till 3% and then decreases afterward and for DIFW-sd, cv-score increases till 5% and then drops from there. From Fig. 19, we can observe that the performance of DIFW-fd and DIFW-sd increases very slightly till $\beta$ is 70% and slightly drops afterward. From Fig. 20, we can observe that for the values of $\gamma$ the cv-score of DIFW-fd monotonically increases till 15% and then decreases afterward and for DIFW-sd, cv-score increases till 20% and then drops from there.

We have conducted experiments for other datasets and found that the best performances of DIFW-fd and DIFW-sd are found when $\alpha$, $\beta$, and $\gamma$ are approximately at 3%, 70%, and 15%, respectively. Based on this observation, we used the same values as the default values in all the preceding experiments.

## 5 Conclusions and future work

In this paper, we propose an improved framework to represent a document using the word embeddings. The existing document representation models represent the document in the same feature space as that of word embeddings, and they are based on vector averaging. As a result, they suffer from the natural drawbacks of averaging. In the proposed novel document representation framework, a document is modeled as a vector of distances from multiple words in a different higher-dimensional feature space. We proposed two methods for the selection of potential feature words and presented a distance function to measure the distance between the feature word and the document. We empirically evaluated these feature selection approaches and the distance measure. Experimental results on multiple data sets demonstrate that the proposed framework improves the classification accuracy significantly as compared to the baseline methods.

The proposed model is simple and represents the document by capturing the multiple aspects of the document in an effective manner. Also, the proposed approach represents a document with words as features which are interpretable. Overall, the proposed model provides an alternative framework to represent the larger text units with word embeddings and provides the scope to develop new approaches to improve the performance of document-based applications.

As a part of future work, we are planning to extend the proposed model to investigate approaches to improve the performance of other natural language processing tasks. The word frequency and the radius are the criteria for the proposed feature words selection approaches. We are planning to combine both the selection criteria to employ a hybrid feature words selection approach. Also, we are planning to develop a word weighting scheme using both frequency and radius to improve the performance.

## References

1. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: International Conference on Learning Representations (2017)
2. Baroni, M.: 39 distributions in text. In: Corpus Linguistics: An International Handbook, vol. 2, pp. 803–822 (2005)
3. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
4. Blashfield, R.K.: Mixture model tests of cluster analysis: accuracy of four agglomerative hierarchical methods. Psychol. Bull. **83**(3), 377 (1976)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
6. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
7. Camacho-Collados, J., Pilehvar, T.: From word to sense embeddings: a survey on vector representations of meaning. arXiv preprint arXiv:1805.04032 (2018)
8. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 160–167. ACM (2008)
9. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
10. De Boom, C., Van Canneyt, S., Demeester, T., Dhoedt, B.: Representation learning for very short texts using weighted word embedding aggregation. Pattern Recognit. Lett. **80**, 150–156 (2016)
11. Dubuisson, M.P., Jain, A.K.: A modified Hausdorff distance for object matching. In: Proceedings of 12th International Conference on Pattern Recognition, pp. 566–568. IEEE (1994)
12. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning, vol. 1, no. 10, pp. 18–20 (2001)
13. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation". AI Mag. **38**(3), 50–57 (2017)
14. Harris, Z.S.: Distributional structure. Word **10**(2–3), 146–162 (1954)
15. Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J.: Comparing images using the Hausdorff distance. IEEE Trans. Pattern Anal. Mach. Intell. **15**(9), 850–863 (1993)
16. Iacobacci, I., Pilehvar, M.T., Navigli, R.: SensEmbed: learning sense embeddings for word and relational similarity. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, pp. 95–105 (2015)
17. Inc., A.E.: Calculating the Average of Averages. https://help.analyticsedge.com/howto/calculating-the-average-of-averages/ (2014). Accessed 20 Aug 2019
18. Johnson, S.C.: Hierarchical clustering schemes. Psychometrika **32**(3), 241–254 (1967)
19. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
20. Kim, H.K., Kim, H., Cho, S.: Bag-of-concepts: comprehending document representation through clustering words in distributed representation. Neurocomputing **266**, 336–352 (2017)
21. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
22. Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: Advances in Neural Information Processing Systems, pp. 3294–3302 (2015)
23. Lance, G.N., Williams, W.T.: A general theory of classificatory sorting strategies: 1. Hierarchical systems. Comput. J. **9**(4), 373–380 (1967)
24. Lau, J.H., Baldwin, T.: An empirical evaluation of doc2vec with practical insights into document embedding generation. CoRR arXiv:abs/1607.05368 (2016)
25. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 1188–1196 (2014)
26. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Trans. Assoc. Comput. Linguist. **3**, 211–225 (2015)
27. Manning, C., Raghavan, P., Schütze, H.: Introduction to information retrieval. Nat. Lang. Eng. **16**(1), 382 (2010)
28. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
29. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
30. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 13, pp. 746–751 (2013)
31. Nickel, M., Kiela, D.: Poincaré embeddings for learning hierarchical representations. In: Advances in Neural Information Processing Systems, pp. 6338–6347 (2017)
32. Nutanong, S., Jacox, E.H., Samet, H.: An incremental Hausdorff distance calculation algorithm. Proc. VLDB Endow. **4**(8), 506–517 (2011)
33. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
34. Sahlgren, M.: The distributional hypothesis. Ital. J. Disabil. Stud. **20**, 33–53 (2008)
35. Salton, G., Yang, C.S.: On the specification of term values in automatic indexing. J. Doc. **29**(4), 351–372 (1973)
36. Sneath, P.H.: The application of computers to taxonomy. Microbiology **17**(1), 201–226 (1957)

37. Sokal, R., Sneath, P.: Principles Numerical Taxonomy, vol. 359. WH Friedman and Company, San Francisco (1963)
38. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. J. Doc. **28**(1), 11–21 (1972)
39. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: an open multilingual graph of general knowledge. (2017)
40. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems, pp. 649–657 (2015)
41. Zipf, G.K.: The Psycho-biology of Language: An Introduction to Dynamic Philology. Routledge, Abingdon (2013)