Text Similarity Function Based on Word Embeddings for Short Text Analysis

Adrián Jiménez Pascual[1](✉) and Sumio Fujita[2]

[1] The University of Tokyo, 3-8-1, Komaba, Meguro, Tokyo, Japan
adri@ms.u-tokyo.ac.jp
[2] Yahoo Japan Corporation, 1-3, Kioicho, Chiyoda-ku, Tokyo, Japan
sufujita@yahoo-corp.jp

Abstract. We present the Contextual Specificity Similarity (CSS) measure, a new document similarity measure based on word embeddings and inverse document frequency. The idea behind the CSS measure is to score higher the documents that include words with close embeddings and frequency of usage. This paper provides a comparison with several methods of text classification, which will evince the accuracy and utility of CSS in k-nearest neighbour classification tasks for short texts.

We experimentally confirmed that CSS performed excellent in the short text classification task as have been intended, outperforming traditional methods as well as WMD, the most recently proposed method.

1 Introduction

One of the most broadly used representations of text documents are bags of words (BOW) weighted by term frequency-inverse document frequency (TF-IDF). Nevertheless, some undesired results can arise specially when using these traditional representations to analyze short texts, as in the following example:

$$\begin{aligned} d_1 &= The\,man\,walked\,into\,the\,bar\ , \\ d_2 &= He\,entered\,a\,pub\ . \end{aligned} \tag{1}$$

These two sentences express the same action with almost synonymous words, yet when considering their BOW representation ρ under the basis [*the*, *man*, *walked*, *into*, *bar*, *he*, *entered*, *a*, *pub*], they become transversal (i.e., unrelated):

$$\begin{aligned} \rho(d_1) &= [2, 1, 1, 1, 1, 0, 0, 0, 0]\ , \\ \rho(d_2) &= [0, 0, 0, 0, 0, 1, 1, 1, 1]\ . \end{aligned} \tag{2}$$

This is an extreme case of the real problem which is the almost-perpendicularity of closely related short texts that use different terminology.

This work was carried out while the first author was in a research internship at Yahoo! JAPAN Research.

A. Gelbukh (Ed.): CICLing 2017, LNCS 10761, pp. 391–402, 2018.
https://doi.org/10.1007/978-3-319-77113-7_31

Many methods have been developed in order to tackle this problem [1,2]. Kusner et al. proposed an interesting distance in this direction called Word Mover's Distance (WMD) as analogy of the Earth Mover's Distance [3], translating an area transfer problem into a word transfer problem. The approach of WMD towards short texts distances served as inspiration for our work, which looks for a word transfer in a simpler and broader meaning than WMD, yet with better results in classification tasks – as will be seen in Sect. 4.

In this paper we provide a new document similarity measure based on the remarkable word embedding model by Mikolov et al. (2013) *word2vec* [4], which was proven by the authors to construct embedded word vectors that preserve semantic relationships when operated. For example, we could consider the following operation: $v(king)$ - $v(man) + v(woman)$, which will result into a vector closest to $v(queen)$. We will represent text documents as arrays of their word vectors, and then make use of this property together with the document's words IDF to define our closeness measure Contextual Specificity Similarity (hereon *CSS*) between documents.

The rest of the paper is organized as follows: Sect. 2 overviews the previous studies of related domains and Sect. 3 explains our proposed similarity measure. In Sect. 4, we presented our evaluation experiments and analyzed the results. Finally, Sect. 5 concludes the paper.

2 Related Work

Computing textual similarity is of great interest not only for natural language processing society but for many related areas including document retrieval [5,6], text classification [7,8], news categorization and clustering [1,9], song identification [10], sentiment analysis [11], and multilingual document matching [12].

2.1 Text Similarity Measures in Information Retrieval

In information retrieval, the task consists of identifying relevant text documents of various length given the description of search requests typically in very short textual query such as TREC topic descriptions [13]. Given the representations of bags of words of both the query and documents, the vector space model computes the similarity between two vectors, each element of which is weighted by TF-IDF, local and global corpus statistics based on term frequencies [5]. More sophisticated text similarity measures based on bags of words include OKAPI BM25 TF [6], which approximates 2-poisson model term weighting and several language modeling approaches [14].

2.2 Context Vectors and Dimensional Reduction Approaches

The history of the discovery of word classes based on contextual information is as old as we may go back to the work of structural linguists in the middle of the 20th century [15]. The origins of several distributional word representations seem to be an inversion of bag of words representation of documents, where a word is represented by the centroid of vectors representing the textual contexts of the appearances [16]. On the other hands, statistical dimensional reduction approaches of document representations, initiated by *Latent semantic indexing* (LSA)[17], try to represent documents by a fewer dimension than the vocabulary size in order to solve word miss matching issues in several text matching applications. Recently, the most successful example is *Latent Dirichlet Allocation* (LDA) by Blei et al. [18], which learns *topic models* consisting of contextually related words by completely unsupervised manner.

2.3 Word Embedding

Finally, we adopted *word2vec*, continuous vector representations of words from very large corpora where the words occur, using a continuous skip-gram model. The neural network learning process is enabled by adopting the negative sampling method which approximately maximizes a softmax objective function of the probability of observing context words given the target word [4,19].

Kusner et al. proposed a document distance measure, WMD on the basis of word embedding representation of words and short texts [3], translating an area transfer problem into a word transfer problem.

3 Contextual Specificity Similarity Measure

Our purpose is to create a method that reckons meaning-related texts that use different terms (i.e., they are unrelated through BOW, as in (1)).

3.1 Background

As previously stated, Kusner et al.'s WMD idea was taken as a starting point, from which our development subsequently diverged.

Formally, WMD's original definition lies on an interpretation of the Earth Mover's Distance, transforming this earth moving problem into a word moving problem. On its basis, they assume that a sentence (area) can be thought of as a certain disposition of words (earth), and finding the similarity of two sentences would equal to minimizing the amount of work one has to do to transport all words in one sentence into the words of the second one.

Technically, we assume we have a word embedding matrix $X \in \mathbb{R}^{d \times n}$, where d is the dimensionality of the word vectors, and n the number of words in the vocabulary corpus. Each element $x_i \in \mathbb{R}^d$ is a vector that represents the i^{th} word in the d-dimensional space. Let $c(i,j) = ||x_i - x_j||_2$ be the "cost" associated to travelling from the i^{th} word to the j^{th} word, and let $T \in \mathbb{R}^{n \times n}$ be a flow matrix whose ij-term represents the "amount of the i^{th} word that travels to the j^{th} word". Then, the problem of calculating the distance between two documents d_1 and d_2 is summed up in the formula:

$$\text{WMD}(d_1, d_2) = \min_{T \geq 0} \sum_{i,j=1}^{n} T_{ij} c(i,j) \ , \tag{3}$$

where certain constrains regarding the matrix T apply with respect to the words from the documents used.

Nevertheless, the WMD distance features a special assignment if the sentences to be compared have different number of words, in which case words are "weight-wise split" and one certain word may be divided into several portions, each of them being transformed into a different term in the comparing sentence. As the authors state, if we consider the sentences "*The President greets the press in Chicago*" and "*Obama speaks in Illinois*", we will get split associations such as "*Obama*" being transported to "*President*" and "*greets*", or "*speaks*" being moved to "*President*" and "*press*". This ambiguity in the assignment of target is what we tried to avoid in our method, which only assigns one word per word.

3.2 Definition

The idea behind our method also lies on a word-weight transfer. However, instead of performing a word-weight transfer from one word to all its "close" words, we only look for one closest word in terms of word embeddings and IDF. We achieve this by creating a word similarity matrix—whose entries will be defined as the product of the average IDF of the facing words and the *cosine similarity* of their vectors—and looking for the maximal values in it.

The reason for taking a matrix with such values as reference is because we consider that for two words to be similar they should have similar word embeddings (i.e., contextual similarity) and we would like higher IDF terms to contribute more in the weighting of the measure, since usually less frequency of occurrence is related to higher specificity of the terms, and this is specially valuable in short texts analysis. For this, we call our method *Contextual Specificity Similarity* (*CSS*).

Thereby, despite *President*, *Prime Minister* and *Churchill* having almost equal word embeddings due to their appearance in similar contexts, the entry in the matrix corresponding to *Prime Minister* versus *Churchill* will show a bigger value thanks to their greater IDF, and therefore they would cast a bigger correspondence value than the pairs *President-Prime Minister* or *President-Churchill.* By doing this, we emphasize the focus of our similarity search on higher IDF words, as mentioned above.

Unlike most other methods, our method assigns higher values to closer words, potentially reaching a maximum when a word is compared to itself (proper distances would become 0 in this case). Therefore, instead of a minimizing function we require a maximizing one.

Similarly to WMD's technical definition, in our construction we assume we are given a word embedding matrix $V \in \mathbb{R}^{d \times n}$ (coming from *word2vec*), where n is the number of vectors (i.e., the number of unique words in our corpus) and d is their embedding dimension. As previously, we consider the vectors to be read in columns. Explicitly, the i^{th} column of V, $v_i \in \mathbb{R}^d$, represents the d-dimensional embedding of the i^{th} word in the corpus. Consider now $\sigma : D \rightarrow \mathbb{R}^d$ to be a function from a document D to the vector space $\mathbb{R}^d$ that assigns to every word $w \in D$ its correspondent vector in V, $v = \sigma(w) \in \mathbb{R}^d$.

Words Similarity Matrix. Let w_i and w_j be two words whose word embedding vectors are $v_i = \sigma(w_i)$ and $v_j = \sigma(w_j)$ respectively. We first define the following matrix:

$$M(w_i, w_j) = \frac{v_i \cdot v_j}{||v_i||||v_j||} \cdot \frac{IDF(w_i) + IDF(w_j)}{2}\ , \tag{4}$$

where *IDF* is the *inverse document frequency* of the words through the all documents. Expressed in words, each entry of the matrix M is the *cosine similarity*[1] of the vectors associated to the words, weighted by the average of their IDF. This means that the closer two words are in the embedding, and the less frequent they are in appearances, the higher the assigned value will be. Please, observe that the diagonal of the matrix represents the IDF of all words: $M(w_i, w_i) = IDF(w_i)$.

Document Similarity Measure. Having created this word-similarity matrix M, we now define the similarity between documents as:

$$CSS(d_1, d_2) = \sum_{w_1 \in d_1} \max_{w_2 \in d_2} M(w_1, w_2)\ . \tag{5}$$

By defining *CSS* in this manner, for every word w_1 in d_1, we look for the word in d_2 with the highest similarity to it. We do this for every word in d_1, therefore at the end we are adding the values of all most-similar words to the words of d_1 in d_2.This contrasts with the definition of WMD, with which one word can be transformed ("moved") into several words, while our method converts one word into the most similar it finds under these requisites.

It is important to note that, actually, the CSS measure is not a formal *distance*, since neither the properties $d(a, a) = 0$ or $d(a, b) = d(b, a)$ are satisfied in general. Nevertheless, this closer-contextuality-higher-value measure will be proven to be an effective measure for the problem that matters.

Example 1. Let us illustrate how CSS works with a basic example where traditional BOW based methods would fail to grasp texts similarities:

$$\begin{aligned} d_1 =& \mathit{Child\,of\,mine}\ , \\ d_2 =& \mathit{Mother\,of\,his}\ , \end{aligned} \tag{6}$$

$$M = \begin{bmatrix} 2.2126 & 0.1777 & 1.4609 & \mathbf{1.9512} & 1.1209 \\ 0.1777 & 0.0333 & 0.2363 & \mathbf{0.2419} & -0.0629 \\ 1.4609 & \mathit{0.2363} & 3.0019 & 1.0636 & \mathbf{1.1829} \\ \mathit{1.9512} & 0.2419 & 1.0636 & 3.8312 & 1.6852 \\ 1.1209 & -0.0629 & \mathit{1.1829} & 1.6852 & 3.7512 \end{bmatrix}\ . \tag{7}$$

[1] In practice we will implement the cosine similarity as the dot product without normalization, since the word vectors obtained from *word2vec* have a modulus close to 1, and making the whole calculation would increase the complexity to the algorithm while not improving the results.

The matrix M is expressed in the basis $[w_1, w_2, w_3, w_4, w_5] =$ [*mine, of, Child, his, Mother*]. The similarity measure between d_1 and d_2 will therefore be:

$$\begin{aligned} CSS(d_1, d_2) =& M(w_1, w_4) + M(w_2, w_4) + M(w_3, w_5) \ , \\ CSS(d_1, d_2) =& 3.3760 \ , \end{aligned} \tag{8}$$

where the maximal terms are **bold** in the matrix. Remember that this similarity measure is not symmetrical. As mentioned before, observe that:

$$\begin{aligned} CSS(d_2, d_1) =& M(w_4, w_1) + M(w_2, w_3) + M(w_5, w_3) \ , \\ CSS(d_2, d_1) =& 3.3704 \ , \end{aligned} \tag{9}$$

which is different from $CSS(d_1, d_2)$. The maximal terms of M when calculating the similarity measure from d_2 towards d_1 are *italized* in the matrix.

Extending the example, if we added a third sentence $d_3 =$ *Colors of signs* to Eq. 6 (what gives us a bigger M), and calculated its similarity with the previous sentences, we would get:

$$\begin{aligned} CSS(d_1, d_2) &= 3.3760 \ , \\ CSS(d_1, d_3) &= 1.0927 \ , \\ CSS(d_2, d_3) &= 0.8199 \ . \end{aligned} \tag{10}$$

These values go along with the idea of similarity with which we defined the measure.

3.3 Derived Document Similarity

In addition to the definition of CSS, we define yet another similarity measure based on it but with a slight change that improves its definition when targeted to long texts.

The basic approach remains the same: the similarity between words (i.e., the matrix M) is as previously defined. However, in this occasion instead of simply adding the similarity value of the word with most similar features for a given word, we will only count with the values of those words whose reciprocal corresponds to itself, and then take their average similarity value. In other words, if we have $w_1 \in d_1$ and $w_2 \in d_2$ such that w_2 is the most similar term to w_1 in d_2, we will count their (weighted) similarity value if and only if w_1 is the respective most similar term to w_2 among all the words in d_1.

We consider this new approach to emphasize the resemblance between sentences, since now only pairs of similar terms will contribute to the summation.

Definition. For the technical details of this similarity measure, consider M to be defined as in Eq. 4. Using the same notation as before, let us first define the following set:

$$\mathcal{A}(w, d) := \{w' \in d | M(w, w') = \max_{w_i \in d} M(w, w_i)\} \ . \tag{11}$$

$\mathcal{A}(w,d) \subset d$ is the set of words in the document d closest to the word w. This set would generally consist of one single word, and we shall assume so in the successive part. Consider now that the word w_1 belongs to the document d_1, and let d_2 be another document. Then, we define the following *delta* function:

$$\delta(w_1, d_1, d_2) := \begin{cases} 1 & \text{if } w_1 = \mathcal{A}(\mathcal{A}(w_1, d_2), d_1) \ , \\ 0 & \text{else } \ . \end{cases} \tag{12}$$

This function becomes 1 only when w_1 is the associated word corresponding to the associated word of itself. Equivalently, if $w_2 \in \mathcal{A}(w_1, d_2)$, then $\delta = 1$ only if $w_1 \in \mathcal{A}(w_2, d_1)$.

We now define the set of associated pairs between d_1 and d_2 as:

$$\mathcal{P}(d_1, d_2) := \{(w_1, \mathcal{A}(w_1, d_2)) \in d_1 \times d_2 | \delta(w_1, d_1, d_2) = 1\} \ . \tag{13}$$

Using this, we define our new measure:

$$CSS{*}(d_1, d_2) = \frac{1}{|\mathcal{P}(d_1, d_2)|} \sum_{(w_1, w_2) \in \mathcal{P}(d_1, d_2)} M(w_1, w_2) \ . \tag{14}$$

This new measure that we will call *CSS**, represents a weighted modification of CSS. The summation of CSS* partialy realizes the summation of CSS, since it only takes into account the summands when they are symmetrical in the sense of δ. The result is then averaged by the amount of terms that were actually summed, what gives us a mean value of the relevant word similarities.

This definition extracts the similarity between documents based on their reciprocal word similarity. The longer the documents are, the less likely it is to find a proper pair, yet the more precise the match when found.

Example 2. In *Example* 1 we saw that the closest word to "*Child*" (w_1) was "*Mother*" (w_4), and the closest word to "*Mother*" was "*Child*". So happened too with "*mine*" (w_3) and "*his*" (w_5). But we find that despite "*Mother*" (w_4) being the closest word to "*of*" (w_2), "*of*" is <u>not</u> the closest word to "*Mother*" (it is "*Child*", as we already said). Therefore, in this scenario, we will only take into account the first couple of words, which are the "corresponded" ones, and the measure would result into:

$$\begin{aligned} CSS{*}(d_1, d_2) &= \frac{1}{2}\big(M(w_1, w_4) + M(w_3, w_5)\big) \ , \\ CSS{*}(d_2, d_1) &= \frac{1}{2}\big(M(w_4, w_1) + M(w_5, w_3)\big) \ . \end{aligned} \tag{15}$$

Please observe that this derived measure is actually symmetrical, since we only add the values if there is reciprocity in similarity terms.

3.4 Other Attempts

These two similarity measures (CSS and CSS*) were chosen after several attempts to design an adequate similarity measure for short text analysis. In particular, our mayor concern was to create a good distance matrix M (Eq. 4), since the definitions of CSS (Eq. 5) and CSS* (Eq. 14) arise quite logically considering what our goal is. Therefore we tried many variations of definition for such M. Specifically, we tested variations on the multiplicand in Eq. 4, since we thought the cosine similarity multiplier ought to remain unchanged to properly reflect a similarity feature between word embeddings.

Among the changes in definition that we performed and whose effect on the final result we compared, we tried taking the minima and maxima of the IDFs respectively instead of the finally chosen average expression:

$$M_1(w_i, w_j) = cos - sim(w_i, w_j) \cdot \min(IDF(w_i), IDF(w_j)) \ , \tag{16}$$

$$M_2(w_i, w_j) = cos - sim(w_i, w_j) \cdot \max(IDF(w_i), IDF(w_j)) \ . \tag{17}$$

None of these led to overall better results. Neither did considering other quantities such as the geometric mean:

$$M_3(w_i, w_j) = cos - sim(w_i, w_j) \cdot \sqrt{IDF(w_i) \cdot IDF(w_j)} \ , \tag{18}$$

or the harmonic mean:

$$M_4(w_i, w_j) = cos - sim(w_i, w_j) \cdot 2 \cdot \frac{IDF(w_i) \cdot IDF(w_j)}{IDF(w_i) + IDF(w_j)} \ . \tag{19}$$

We tried several other arrangements and formulae without further improvements. Nonetheless, we found a pattern which tends to improve the results for every modification that we tried: squaring the matrix (element-wise) –remember that for our usage of similarity, the bigger the value the higher the similarity.

$$M'_*(w_i, w_j) = M_*(w_i, w_j)^2 \tag{20}$$

This would lead to a better performance than the non-squared case in most cases. However, we decided not to stick to this method due to our uncertainty of a plausible explanation for this effect.

4 Evaluations

4.1 Evaluation Environment

The test is run through two sets of $1,000$ and $10,000$ Japanese articles from Mainichi-shimbun documents in NTCIR-3 data [20] respectively classified with a section tag within the newspaper (*culture, sports, politics,* etc.). These articles are in turn split in their titles and bodies. Beside CSS and CSS*, we run the test using some classical retrieval methods (BM25 and TF-IDF) by Terrier

IR platform[2], a random classifier, and, for the sake of comparison, the WMD distance[3].

For all methods, the modus operandi is:

1. For every document (title or body) $d \in D$, calculate its distance to all remaining documents in the corpus $d_i \in D \setminus \{d\}$.
2. Rank d to all d_i's distances in closeness order (for CSS and CSS* closer are larger values).
3. Determine the class to which d should belong by utilizing the k-nearest neighbours (k-NN) method, applied with $k = 1$, $k = 5$ and $k = 15$.
4. Results are presented in terms of the macro-average of the *F-measure* of each section tag, where the F-measure is calculated as usual as the harmonic mean of the evaluation *precission* and *recall*:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{21}$$

The word embedding used is the one provided by *word2vec*, where the vectors have been trained over more than 63 million words, spread through 220 thousand articles: 3+ million words in titles (15 words each in average), 60+ million words in bodies (275 words each in average).

4.2 Results

The results are presented in the four tables below, which are divided in the analysis of 1,000 documents on the left column and 10,000 documents on the right column, and row-wise the analysis of the body of the articles above and their title below. Maximum values are highlighted in **bold** (Tables 1, 2, 3, and 4).

Table 1. Body – 1,000 documents

Method	$k = 1$	$k = 5$	$k = 15$
BM25	**0.4399**	0.3495	0.2904
TF-IDF	0.4346	**0.3539**	0.2863
WMD	0.3151	0.2397	0.2182
Random	0.0447	0.0578	0.0478
CSS	0.3618	0.2598	0.1946
CSS*	0.4214	0.3382	**0.3026**

Table 2. Body – 10,000 documents

Method	$k = 1$	$k = 5$	$k = 15$
BM25	**0.5135**	**0.4815**	**0.4706**
TF-IDF	0.5107	0.4786	0.4675
WMD	—	—	—
Random	0.0638	0.0568	0.0486
CSS	0.3315	0.2783	0.2569
CSS*	0.3551	0.3035	0.2926

[2] http://terrier.org/.

[3] Due to calculations limits (memory error), the WMD distance was only calculated for the set of 1,000 articles.

Table 3. Title – 1,000 documents

Method	$k = 1$	$k = 5$	$k = 15$
BM25	0.4862	0.4591	0.3790
TF-IDF	0.4848	0.4537	0.3650
WMD	0.1322	0.1001	0.0871
Random	0.0558	0.0485	0.0472
CSS	**0.5332**	**0.4713**	**0.4151**
CSS*	0.4432	0.3829	0.3417

Table 4. Title – 10,000 documents

Method	$k = 1$	$k = 5$	$k = 15$
BM25	0.6396	0.6508	0.6127
TF-IDF	0.6433	0.6554	0.6167
WMD	—	—	—
Random	0.0566	0.0541	0.0471
CSS	**0.6523**	**0.6651**	**0.6326**
CSS*	0.4006	0.3947	0.3917

4.3 Discussions

The effectiveness of our method is clearly reflected in the two lower tables, which show how CSS outperforms any other contrasted method in short text classification tasks using the k-nearest neighbours method. Whilst on longer texts, both CSS and CSS* are overwhelmed by more broadly used methods such as BM25 or TF-IDF. Yet, as expected, CSS* shows a better performance in longer texts analysis than CSS, what supports the motivation behind CSS*.

As for WMD, which served as inspiration for developing our first method, it does not show a good performance specially in short texts analysis, what could be due to the lack of a broader background context that could help words find better pairings. Nonetheless, it performs at similar levels to CSS in long text classification tasks, as it can be seen in Table 1. Unlike the results that Kusner et al.'s paper [3] reported, BM25 as well as TF-IDF performed better in longer text as have been proven in the series of past evaluation forums in information retrieval [13]. One reason of such overwhelming performance of traditional approaches is that we used Terrier IR platform implementation for TF-IDF and BM25, which is properly configured at the out of box status. As these methods leverage local as well as global statistics, a carefully configured corpus setting and operational parameter setting are needed to be well performed; failing to do that leads to a very weak baseline performance.

In spite of such strong baselines, we can briefly summarize that CSS performs especially excellent in short text classification as have been intended, outperforming traditional methods such as TF-IDF and BM25 as well as WMD, the most recently proposed method. Although CSS* is fairly good in long text classification, traditional methods such as TF-IDF or BM25 performed much better when properly configured.

5 Conclusions

We proposed two text similarity measures, namely CSS and CSS*, among which CSS is intended to improve the effectiveness in short text matching where word miss matching is a crucial problem. According to our experiments described in Sect. 4, we can conclude that CSS and CSS* are powerful tools for short and long text classification tasks respectively, being CSS the best classifier among the compared methods for short texts. Especially CSS showed excellent performance in short text classification as have been intended, outperforming traditional methods such as TF-IDF and BM25 as well as WMD.

References

1. Greene, D., Cunningham, P.: Practical solutions to the problem of diagonal dominance in kernel document clustering. In: Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, 25–29 June 2006, pp. 377–384 (2006)
2. Schölkopf, B., Weston, J., Eskin, E., Leslie, C., Noble, W.S.: A kernel approach for learning from almost orthogonal patterns. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, pp. 511–528. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36755-1_44
3. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From word embeddings to document distances. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015, pp. 957–966 (2015)
4. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013)
5. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **24**, 513–523 (1988)
6. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Croft, B.W., van Rijsbergen, C.J. (eds.) SIGIR 1994, pp. 232–241. Springer, London (1994). https://doi.org/10.1007/978-1-4471-2099-5_24
7. Lewis, D.D.: An evaluation of phrasal and clustered representations on a text categorization task. In: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1992, pp. 37–50. ACM, New York (1992)
8. Lewis, D.D.: Evaluating and optimizing autonomous text classification systems. In: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1995, pp. 246–254. ACM, New York (1995)
9. Ontrup, J., Ritter, H.J.: Hyperbolic self-organizing maps for semantic navigation. In: Advances in Neural Information Processing Systems 14, Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, 3–8 December 2001, Vancouver, British Columbia, Canada, pp. 1417–1424 (2001)
10. Brochu, E., de Freitas, N.: "Name that song!" A probabilistic approach to querying on music and text. In: Advances in Neural Information Processing Systems 15, Neural Information Processing Systems, NIPS 2002, 9–14 December 2002, Vancouver, British Columbia, Canada, pp. 1505–1512 (2002)
11. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retrieval **2**, 1–135 (2007)
12. Quadrianto, N., Smola, A.J., Song, L., Tuytelaars, T.: Kernelized sorting. IEEE Trans. Pattern Anal. Mach. Intell. **32**, 1809–1821 (2010)
13. Harman, D.: Overview of the first TREC conference. In: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1993, pp. 36–47. ACM, New York (1993)

14. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. **22**, 179–214 (2004)
15. Harris, Z.: Structual Linguistics. University of Chicago Press, Chicago (1951)
16. Schütze, H.: Automatic word sense discrimination. Comput. Linguist. **24**, 97–123 (1998)
17. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**, 391–407 (1990)
18. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
19. 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 26: 27th Annual Conference on Neural Information Processing Systems 2013, Proceedings of a meeting held 5–8 December 2013, Lake Tahoe, Nevada, United States, pp. 3111–3119 (2013)
20. Chen, K., et al.: Overview of CLIR task at the third NTCIR workshop. In: Proceedings of the Third NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering, NTCIR-3, Tokyo, Japan, 8–10 October 2002 (2002)