Journal of Visual Languages and Computing 48 (2018) 178–186

Contents lists available at ScienceDirect

Journal of Visual Languages and Computing

journal homepage: www.elsevier.com/locate/jvlc

Visual exploration and comparison of word embeddings

Juntian Chen, Yubo Tao*, Hai Lin

State Key Lab of CAD&CG, Zhejiang University, China

ARTICLE INFO

Keywords:
Visual comparison
Word embeddings

ABSTRACT

Word embeddings are distributed representations for natural language words, and have been wildly used in many natural language processing tasks. The word embedding space contains local clusters with semantically similar words and meaningful directions, such as the analogy. However, there are different training algorithms and text corpora, which both have a different impact on the generated word embeddings. In this paper, we propose a visual analytics system to visually explore and compare word embeddings trained by different algorithms and corpora. The word embedding spaces are compared from three aspects, i.e., local clusters, semantic directions and diachronic changes, to understand the similarity and differences between word embeddings.

1. Introduction

The word embedding is a kind of mathematical representation of vocabulary. Usually, there are two kinds of representations: one-hot vector representation and distributed representation. One-hot vector representation easily comes to our minds, which uses the index in the dictionary to represent word uniquely. However, this representation method only separates word and does not express the semantic meanings of the word. Distributed representation is a vector of real numbers, originally proposed by Hinton [1] in 1986, and it can encode semantic information compared to the one-hot vector representation. Euclidean distance or cosine similarity between distributed vectors can be used to measure the semantic similarity of words. Due to these advantages, distributed representation is widely used in many natural language processing tasks.

In order to generate the distributed representation of words, Xu and Rudnicky [2] applied neural networks to train word embeddings in 2000. The classic training method is to build a model by a three-layer neural network proposed by Bengio et al. [3], and many subsequent algorithms are based on this work. Until 2013, Mikolov et al. [4] proposed CBOW and Skip-gram methods. Google made it as an open source project named word2vec based on these methods and made word embeddings widely accepted by users. Thus, we use word2vec to train our word embeddings in this paper. Since the distributed representation captures semantic meanings of words, it is very popular in natural language processing tasks and can significantly improve performance in downstream tasks, such as text classification [5], sentiment analysis [6,7], and semantic analysis [8]. For linguists, word embeddings not only can help them understand the structure of language from a macroscopic perspective, but also analyze the usage and meaning of words in a fine-grained manner.

The original word embedding data is a kind of vector data of tens of dimensions or even hundreds of dimensions. The high-dimensional data is difficult to interpret. It is complicated to show the spatial information in the high-dimensional space. Although we know that semantically similar words would be together, it is still hard to imagine how these words will be distributed in the high-dimensional space. We need to employ visualization techniques to enhance our understanding of the word embedding space. The word embedding space contains semantic information, such as synonym information, and it can be interpreted by the nearest neighbor words or clustering results. It also contains other semantic information, such as the word analogy relationship, and such information requires different visualization techniques to reveal its underlying relationships and structure.

Different corpora, parameters settings and training methods may lead to different word embedding spaces, and they are difficult to evaluate mathematically. The data, what we used to train the word embeddings, is a kind of large text and usually in Gigabytes. In our paper, we use CBOW and Skip-gram method, which are totally different in model architecture, which will discuss in Section 3. Therefore, we hope to design a visual analytics system to explore and compare the similarity and differences between two word embedding spaces trained from different corpora or methods. The contributions of this paper are:

- We propose an interactive visual analytics system to understand and compare the word embedding spaces, in order to obtain an intuitive understanding between the spaces.
- A case study demonstrating insights in word embeddings with

[*] Corresponding author.
E-mail addresses: chenjuntian@zju.edu.cn, jhcjt@vip.qq.com (J. Chen), taoyubo@cad.zju.edu.cn (Y. Tao), lin@cad.zju.edu.cn (H. Lin).

https://doi.org/10.1016/j.jvlc.2018.08.008
Received 20 July 2018; Accepted 20 August 2018
Available online 24 August 2018
1045-926X/ © 2018 Elsevier Ltd. All rights reserved.

different training algorithms and corpora, which reveals some interesting results like latent semantic changes in words.

2. Related work

The word embeddings encode a word as a point in the high-dimensional space. To some extent, the spatial information corresponds to the semantic information of the word. Word embeddings training methods are various and most of them based on statistical language models proposed by Bengio et al. [3]. The most widely used algorithms are Google's word2vec [4] and Glove [9]. Lai et al. [10] outlined lots of word embeddings training methods and provided a number of evaluation criteria. Many methods have been proposed to improve word embeddings for subsequent tasks. Levy et al. [11] focused on transferable hyperparameters in training. They explored how parameter settings impact the quality of word embeddings. Mu et al. [12] demonstrated a post-processing technique to eliminate the common mean vector and a few top directions in word embeddings. Gittens et al. [13] paid their attention to the vector additivity and used sufficient dimensionality reduction framework [14] to optimize the linear vector additivity. Nevertheless, the word embedding training process is very complex and looks like a black box model. Just as the recent popular research domain of deep learning visualization, Rong and Adar [15] attempted to explain why word embeddings make sense through visualizing the training model. In addition, they allowed domain experts to control the training process.

In fact, word embeddings have been applied in many fields. The concept of vectorization is not restricted in words. Kusner et al. [5] used the word vector to calculate the distance of the entire document. Arora et al. [16] took the sentence as a unit to generate a sentence embedding. They inputed a pre-trained word embeddings and proposed an unsupervised method to generate the embeddings of sentences. Apart from natural language processing, in the research field of graphs, Deepwalk [17] migrated the method of word embeddings to nodes in graphs and proposed node embeddings. It achieved good results in some complex tasks, such as overlapped clusters detection.

In text visualization, many research focus on topic visualization and sentiment analysis of text visualization [18]. The topic model is an essential part of text processing and data mining, and it can effectively extract topic information from a large corpora. Alexander and Gleicher [19] proposed a new kind of chart named buddy plots to encode similarity information of models in a single axis. Smith et al. [20] proposed a visual analytics system for hierarchical topic models. Yang et al. [21] proposed an effective analysis system of large documents with hierarchical topic modeling. Xu et al. [6,7] mainly focused on the sentiment analysis of review data, and the analysis of the two extremes of the business commentary against the corpus data of the review. Collins et al. [22] used parallel coordinates to display the corpus and analyze the relations of cases in different courts. A lot of researchers [23–26] built topic models of corpora. Smith et al. [24] extracted the term co-occurrence and covariance of the topic and made a view. In the other respect, Wang et al. [25] proposed a radial hierarchical tree layout algorithm to show the connections and intersections between topics. Liu et al. [27] presented an online visual analytics approach to explore and understand hierarchical topic evolution with text streams. Without using LDA(Latent Dirichlet Allocation), Choo et al. [26] came up with a matrix modeling-based topic modeling and visualization method. Other works focused on the documents, such as Jigsaw [28] and DocuCompass [29]. They paid their attention to the relevance and relations between documents. Jigsaw [28] provides an interactive system to explore latent semantic relations between documents. DocuBurst [30] not only analyzes the documents but also reveals the relationship between words and documents.

For vocabulary visualization, Google has plotted word embeddings generated from word2vec in the 3-dimensional coordinates named Embedding Projector [31]. They trained their corpus by Tensorflow, which is the most popular deep learning framework. Liu et al. [32] designed an interactive visual analytical system to show the word analogies. They focus on the subset of tasks that emphasized visualization of the semantic direction of the word analogy pairs. What they trying to do was using the proposed algorithm to classify the word analogy pair directions and making the directions parallel. ConceptVector [33] applies the word embedding method to model the concept, and the user can customize the semantic concept through interaction and display it in a visual way.

3. Background

Word embeddings have evolved since the concept emerged. The one-hot representation use indexes to represent words but we know little about semantic information. For enriching the meanings of word embeddings, researchers use words frequencies and context information to encode the semantic meaning in word embeddings. Therefore, the distributed representations of words came up. With the popularity of neural network techniques, Mikolov et al. [4] proposed the CBOW and Skip-gram to generate word embeddings.

The input of the algorithms is the same, which is a large text collection from Internet or publications. We also denote the input data as corpora. The basic idea of the algorithm is the Bayesian formula. During the training, we focus on a window, which is a couple of words around the current word, to maximize the probability of the word in the window.

$$\prod_{w\in C} p(w|Context(w)), \tag{1}$$

where C is the corpus and $context(w)$ represents the word w's context word set. After the training, we obtain word embeddings for words in the corpus. In our paper, we use the CBOW and Skip-gram to train word embeddings. The main difference between them is the generation architecture of the estimation. As shown in Fig. 1, CBOW uses context words to predict the current word, but the Skip-gram is the opposite, using the current word to predict context words.

It is obvious that these two algorithms have an apparent difference in model architecture. However, because of the high dimension, the distinction of word embeddings respectively trained by the two algorithms is hard to interpret. As a result, we design several visualization methods to intuitively present the similarity and difference between word embeddings.

4. Tasks

Word embeddings are an essential part of many natural language processing tasks. The quality of word embeddings strongly affects the performance of these tasks. However, the evaluation of word embeddings is challenging and there are no quantitative baseline criteria. In order to explore and compare the semantic properties of the high-dimensional word embedding spaces, we define four tasks after reviewing the literature related to word embeddings in the natural language processing field.

- *Task 1:* Display an overview of a word embedding space. For a word embedding space, we need to intuitively display the structure of word distribution, such as the spatial and semantic clusters. By dimension reduction, we can show clusters of words in terms of word embeddings, the semantic direction, and the relationship between semantic similar words.

- *Task 2:* Analyze the clusters of semantically similar words. As we all know, we can classify words by their meanings. For example, pork, beans, banana, etc. are food terms, school, bookstores, supermarkets, etc. mean place, and fathers, mothers, daughters, and others belong to pronouns. The vocabulary also has its semantic

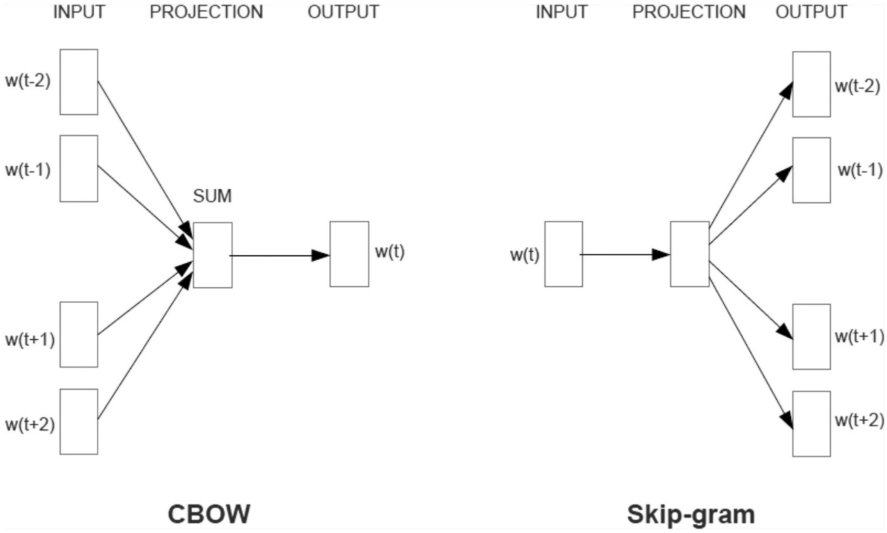

Fig. 1. The model architectures proposed by Mikolov et al. [4]. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

cluster relationship in the language. Therefore, we need to explore the relationship of semantic clusters in the word embedding space, so as to investigate the similarities and differences of the word embedding space, and also can simply evaluate the quality of word embeddings from these clusters.

- *Task 3:* Compare semantic direction. Word pairs often play a very important role in natural language processing and we call word pairs with semantic meanings 'analogy'. One characteristic of distributed word embedding is that they can represent linear semantic directions. For example,

$$w_{king} - w_{queen} \approx w_{man} - w_{woman},\ w \in R^n. \tag{2}$$

The linear semantic directions are highly preserved in the high-dimensional space. We focus our research on the direction of the semantic pair, i.e.

$$d_{king-queen} = w_{queen} - w_{king},\ d \in R^n. \tag{3}$$

We can also compare two word embeddings spaces by comparing the differences in the direction of analogies.

- *Task 4:* Show semantic similar words of a selected word. Neighborhood words that are close to the one word often indirectly reveal semantic information about the meaning of the word. The semantic similar words are also spatially close in the word embedding spaces. For example, the meanings of the word around a polysemy can indicate which meaning of the word in the corpus dominates. Hamilton et al. [34] proposed that the semantic meanings of words will change over periods. They analyzed the semantic information of the words which surround the current word. Linguists can often see the change of semantics or the change of word usage from the analysis of the semantic similar words, which can also provide a supplement for the downstream natural language processing.

5. Visual design

In this section, we introduce our method. First, we define our data format and the alignments algorithm. Then we focus on our visual analytic views.

5.1. Word embedding alignment

5.1.1. Definition

The dictionary is the set of the words:

D = {King, Queen,...}, m=$|D|$.

The high-dimensional word embedding space is defined as:

$S = \{w_1,\ w_2,\ \ldots, w_i,\ \ldots, w_m\}$,

where w_i in R^n, for all i in [1, m]. n is the dimension of word embeddings, usually in 50–500 in the distributed representation. The word embedding of '*king*' is defined as:

$w_{king},\ w \in R^n$.

5.1.2. Layout

This paper uses two algorithms in word2vec: CBOW and Skip-gram. Suppose we use two algorithms to train the corpus separately and generate two word embedding spaces named S_c and S_k. Before visualizing the high-dimensional word embedding space, we need to align the two embedding spaces S_c and S_k, since word embedding algorithms are inherently stochastic and the resulting embeddings are invariant under rotation. Even for the same corpus and training algorithm, two separate training results will produce totally different vectors. Thus, the alignment is necessary for effectively showing the differences between word embeddings. Since the relative positions of many commonly used words are stable, their positions in the aligned word embedding space should be basically the same. Therefore, we employ these words as match points to align two word embedding spaces to identify the word in different positions in the aligned word embedding space. The alignment algorithm uses a linear transformation to preserve the general structure, and it can be transformed into an orthogonal Procrustes problem [34].

The two word embedding sets (S_c and S_k) are first organized into two $m \times n$ real valued matrices A and B, respectively. The ith row of each matrix corresponds to $w_i \in R^n$, and the number of rows is the number of words in the corpora. We then find a $n \times n$ orthogonal matrix Q to minimize the equation:

$$||B - AQ||_F, \tag{4}$$

where $\|\|\|_F$ is Frobenius norm. In this way, we can align two word embeddings sets with the error as little as possible.

The distance between word embeddings can reveal the semantic distance between words. In this paper, the word distance (similarity) is measured by cosine similarity:

$$\cos(w_i, w_j) = \frac{w_i \cdot w_j}{|w_i| \times |w_j|}, \tag{5}$$

where w_i and w_j are the word vectors of two words.

5.2. Clustering view

As mentioned in the task, we expect to explore the semantic information of words. Thus, we design the clustering view to show the similarities and differences between clusters in the high-dimensional word embedding space.

We need to project the high-dimensional word embeddings into a 2D space to show the structure of word distributions and clusters. Since the two word embedding sets are aligned in a common space, we achieve this by t-distributed stochastic neighbor (t-SNE) [35]. One strategy is that two word embedding sets are processed by t-SNE separately. However, t-SNE is a nonlinear dimension reduction method, and this nonlinearity may be different in two dimension reduction processes, which would make nearby words in the common space reduced to different positions in the two cluster views, as shown in Fig. 3. Thus, we adapt the other strategy to reduce two word embedding sets together.We first concatenate two matrixes of two word embedding sets, such as two $m \times n$ matrices A and B of S_c and S_k, into a large matrix, such as a $2m \times n$ matrix $[A^T B^T]^T$, and then reduce all words in the large matrix into a 2D space. In this way, the same nonlinearity is applied to the two word embedding sets, resulting in a more comparable dimension reduction result, as shown in Fig. 4.

In the cluster view, each point represents a word in the word embedding set, and the color encodes the label information if available. The numerical value of the coordinates is just for the relative position comparison. When hovering over a point, the word and its label information will appear nearby the point, and the corresponding cluster will be filled in red. This shows how tight these words in the cluster are, so that the outliers can be found clearly and easily.

From Fig. 4, we intuitively see the difference between the two word embedding sets by comparing the word distribution between clusters, such as the distance between semantic clusters, the compactness of semantic clusters and the relative position of words, to gain a better understanding of the cluster structure in the word embedding space. We can analyze the quality of word embeddings in terms of clusters, such as the closeness of clusters and the distance between clusters (overlap of clusters). This can help users to judge whether the distributed representation in the space is reasonable, so as to explore the cluster relationship of high-dimensional word embedding space.

5.3. Semantic similar word view

As we mention before, the position of the word in the high-dimensional word embedding space corresponds to the semantic information to some extent. In the high-dimensional space, the closer the word's distance (like Euclidean distance) is, the more similar they in word meanings are. Therefore, we use the semantics similar words around the current word (k-nearest neighbors) in the high-dimensional space to express the current word's semantic meaning. The semantic word cloud is designed to show the current word and its semantically similar words in word embeddings, since the word cloud is widely used to show the word set [36].

For the selection of semantic similar words, we use the previously defined cosine similarity to find out the top maximum similarity surrounding words in the word embedding space.

As shown in Fig. 5, we use the force-directed word cloud layout to show neighbors around words. The current word is fixed in the center with the light blue color. The surrounding words represent the top-k semantically similar words related to the current word. The font size of the word represents the cosine similarity between the word and the current word. The larger the font is, the greater the cosine similarity is, the more similar semantics meaning between them. In the layout of the two word clouds, in order to facilitate the comparison of top-k semantically similar words in the word clouds, the same word is placed in the same position of two word clouds with the gray color. Different words are colored in chromatic to emphasize them for visual comparison.

5.4. Analogy view

We propose an analogy view to make the comparison in semantic directions. The semantic direction means two words of the analogy relationship in the high-dimensional space, i.e.

w_i: w_j, $w_i \in R^n$, $w_j \in R^n$.

The direction vector is denoted as d, which is obtained by the difference between two word vectors in the same high-dimensional word embedding space.

$$d = w_i - w_j, \; d \in R^n, \tag{6}$$

where w_i and w_j are the word vector in high-dimensional space. For example, *king-queen* is called an analogy, and its semantic direction is defined in Eq. (3).

The semantic direction is a high-dimensional vector. The semantic directions are first generated in two spaces based on the analogy set, respectively, and then they are reduced to one-dimensional space by using t-SNE, similar to the words in the cluster view. Thus, the direction vector of an analogy becomes a point on the axis. In order to make an intuitive comparison between these semantic directions, the semantic

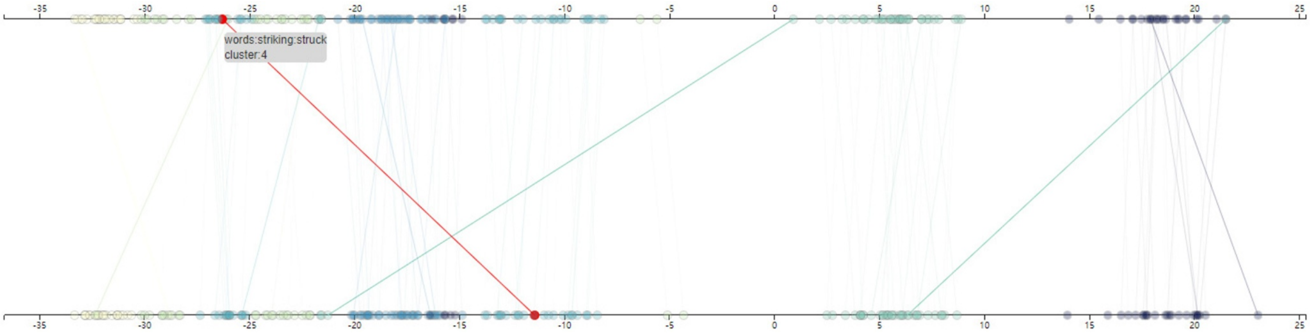

Fig. 2. The analogy view highlights the analogy pair *striking:stuck*. The points in the axes represent the semantic direction reduced into 1-D. The two axes represent two word embedding spaces, respectively. The color of points is the same with the color of clusters, and the slope of each line reveals the difference of the semantic direction in two spaces.

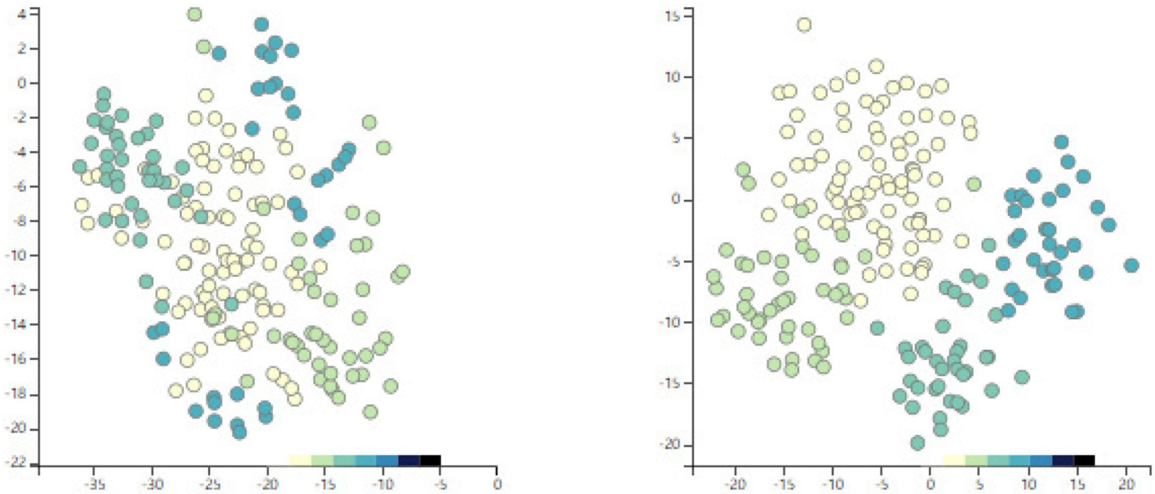

Fig. 3. Two word embedding sets on the wikipedia corpora trained by the CBOW (left) and Skip-gram (right) methods, respectively. The cluster view could not compare these words, when the word embedding sets are not aligned and projected by t-SNE separately.

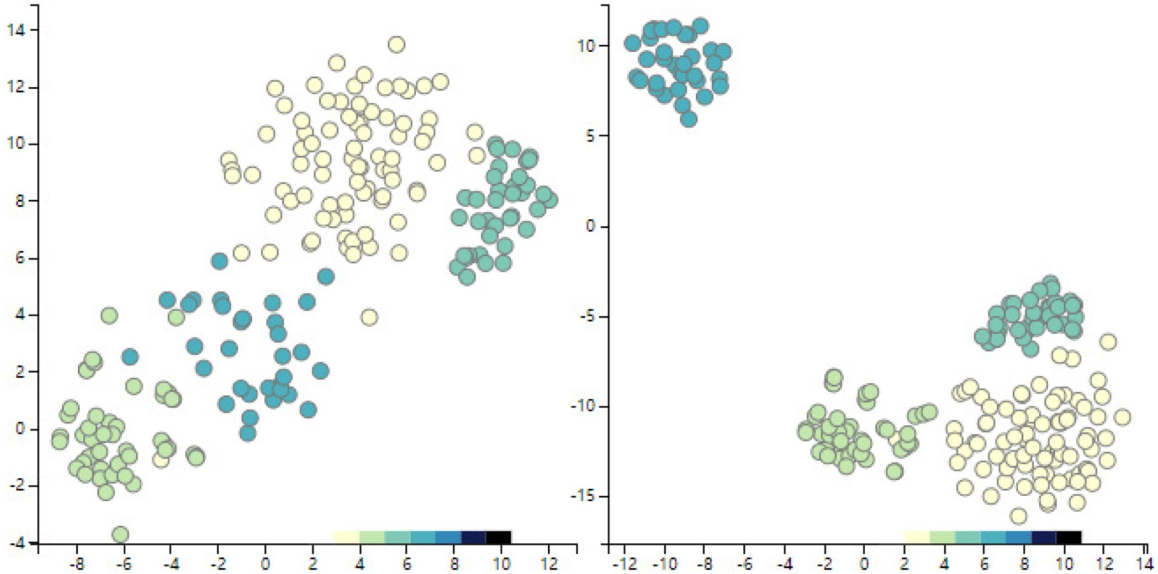

Fig. 4. Two word embedding sets on the wikipedia corpora trained by the CBOW (left) and Skip-gram (right) methods, respectively. When the two word embeddings sets are aligned and reduced together, the cluster view shows more consistent and comparable results.

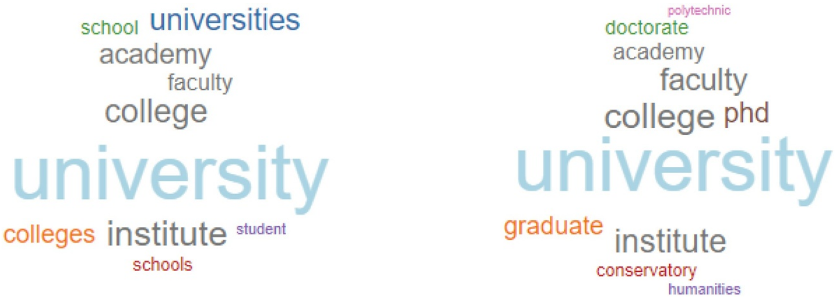

Fig. 5. The semantic similar word view. The current word shows in the center of each part. Each part represents a word embedding space. The surrounding words are the most similar words in the semantic meaning. For comparison, we fix the same words in the gray color in the same place in two word clouds. The other words are colored in chromatic. The font sizes of words encode the similarity. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

directions are split into two sets based on its word embeddings, and represent as points on two parallel axes, respectively. Each point is a semantic direction of an analogy, and two points of the same analogy are connected by a line. Based on the line slope, we can observe the differences between the directions of the analogy in two word embedding spaces. As shown in Fig. 2, if the absolute value of the slope is larger (closer to vertical), it indicates that two semantic directions of the analogy is almost the same in two high-dimensional word embedding spaces. When the absolute value of the slope is smaller (closer to horizontal), it means that two semantic directions of the analogy have a greater difference in the two word embedding spaces. These word pairs, which have a larger difference, deserve our further exploration. In addition, the semantic clusters of analogies are also well displayed on the axes. The word pairs with similar word meanings are close to each other on the one-dimensional axis (indicated by different colors).

The user can select one word pair of interest, and the words of the analogy will be shown near the line. The selected line will be highlighted in the red color, while other lines will have a low transparency as the context. The user can select a word in the cluster view or in the analogy view, and the semantic similar word cloud will show its similar words for exploration and comparison.

6. Case study

6.1. Data preparation

Our corpora come from Wikipedia English corpora in 2017, New York Times corpora from Linguistic Data Consortium (LDC) in 1987 and Yelp comments. We trained word embeddings based on these corpora by word2vec in the genism package. For the training parameters, we set the window size as five and minimum ignoring word count as ten to generate a word embedding set in a 200 dimension. For comparisons, we use the CBOW and Skip-gram algorithm to train corpora respectively with the same parameters.

6.2. Difference caused by training algorithms

We first compare two word embedding training algorithms, CBOW and Skip-gram. The word embedding results are trained from the Wikipedia corpus. We use the nouns in WordNet [37] to analyze the quality of the word embedding sets. These nouns are labeled by categories. We use four types of nouns in the analysis, i.e., food, family, location, and animal. The number of words is quite large. If we visualize the whole vocabulary, the circles may stack and overlap in the cluster view and this makes the view hard to read and interpret. Therefore, we choose some typical words to analyze the local clusters in the word embedding space. As we can see from the clustering view in Fig. 6, where the red color stands for the cluster 4, we can figure out that the word clusters separate into parts after dimension reduction. The word embeddings of Skip-gram (right) shows a better local cluster result in the high-dimensional space than the result of CBOW (left). The clusters separate more distinctively and each cluster shows more compact. Thus, the Skip-gram method may have a better interpretation of local clusters in the word embedding space for the classified nouns in WordNet.

In Fig. 7, we can quickly identify an abnormal point in the animal category cluster. It is far away from other nouns in this category and stays in the green cluster, which means the pronoun. We hover the cursor and see the detail of the word. The word *nanny*, which is prelabeled as animal in WordNet. However, *nanny* actually has the meaning of *female goat*. However we hardly use it with the meaning of *female goat* in the corpus. In the similar word view of Fig. 8, the neighbor words are all with the meaning of *babysitter*. This is a contradiction with the linguistic knowledge in WordNet. In fact, the common usage of *nanny* is the meaning of *babysitter* in the corpus, as the cluster view and semantic similar words view reveals.

6.3. Totally difference in corpus

In order to reflect the impact of different granularities of corpora on the word embedding space, we analyze the influence brought by two totally different corpora. Next, we analyze the impact of changing a bit of corpus in the next case when most of the corpus is the same. The corpora in this case study are the New York Times 1987 corpus and Wikipedia 2017 corpus.

We use eight types of word analogies in Mikolov et al. [4] to compare the semantic relation in two corpora via the analogy view in Fig. 2. We can see that most of semantic directions have no difference, and also show a good clustering result. However, there are several word analogies with different semantic directions, that is the absolute value of the slope is relatively small, even nearly horizontal (that is, the absolute value of the slope is relatively small and horizontal), such as, *new:newer, striking:stuck, generate:generates, bright:brighter, and easy:easier*, whose semantic directions difference in the two spaces are relatively large. These semantic directions are different in the word embedding spaces trained from two different corpora.

In the semantic similar word view, we find that the word *hotel* is more like a functional place in 1987 corpora, which is surrounded by *restaurant, motel, casino*, and *inn*. But in 2017 Wikipedia corpus, the word *hotel* usually comes up with the words *novetel, ramada* and *sheraton*, which mean famous international hotel brands. Nowadays, people see hotel not just a place to sleep but also chase the comfort and enjoyment.

Looking at the word *striking* situation in Fig. 9, we discover the semantic changes through the k-nearest-neighbors at different time steps. In the 1987 corpus, the semantics of the word *striking* is *notable, remarkable,* and *spectacular*, and this is a word with prominent

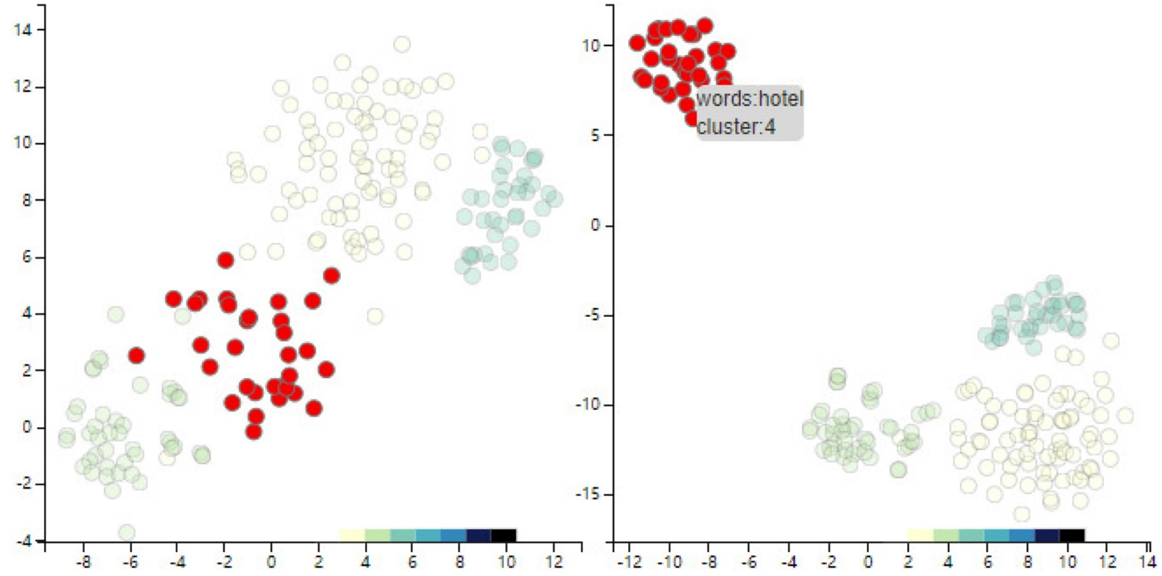

Fig. 6. It shows two different training results of the word *hotel* in the cluster view by CBOW(left) and Skip-gram(right). The word embedding result of Skip-gram shows a better interpretation in the cluster result. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

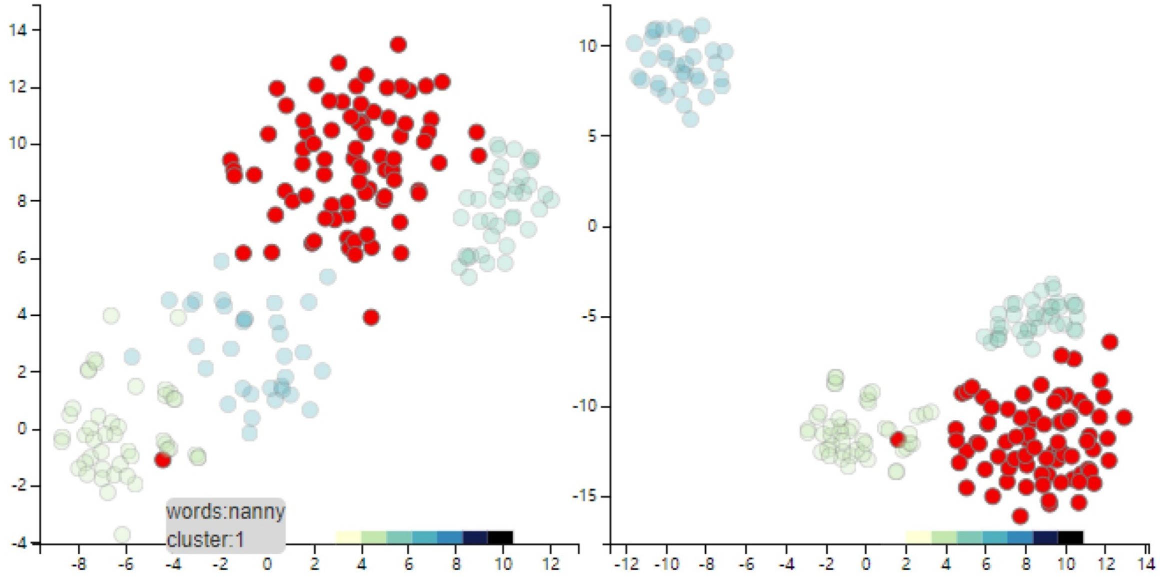

Fig. 7. The word *nanny* in both methods looks like an outlier in the clustering result, which should be labeled in the red cluster (representing animal) but appeared in the green one (representing pronoun). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

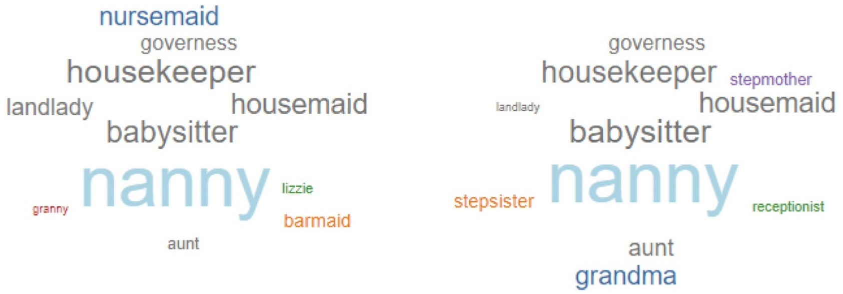

Fig. 8. The semantic similar words around *nanny* all have the meaning of *babysitter*. There is not any word whose meaning is *female goat*.

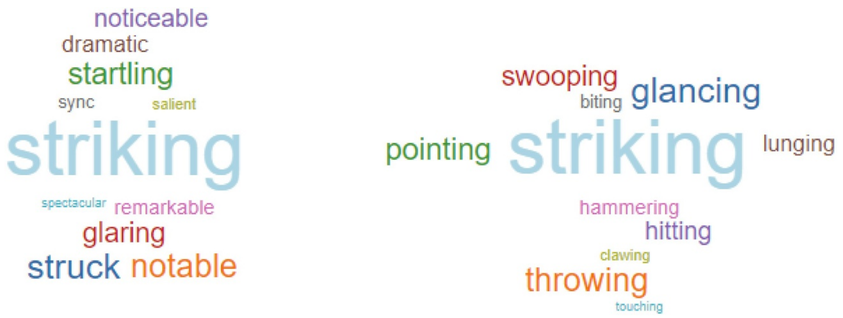

Fig. 9. The latent semantic changes on the word *striking*. The left part shows the meaning in the 1987 New York Times corpus, which meant *outstanding*. However, in the 2017 Wikipedia corpus, the meaning of *striking* becomes *hit*.

significance. In the 2017 corpus, the word *striking* has the majority meaning of *throwing* and *hitting*. We can identify the anomaly in the semantic direction in the analogy view and then find more details in the semantic similar word view for semantic exploration and comparison.

6.4. Fine-tuning on corpus

The influence of corpus on the word embedding space is obvious. The completely different corpus can generate different word embedding sets. We hope discovery that how the corpora affect the word embedding space at a fine granularity, which is to find that a bit of change in linguistic data has an impact in the word embedding space.

For this case study, we add Yelp comment data (1.0 GB) to the 2017 Wikipedia corpus (13.6 GB) and apply the same Skip-gram algorithm and parameters to generate two word embedding sets. Our goal is mainly to explore the subtle influence of the corpus to the word embedding space.

As we can see from Fig. 10, after fine-tuning of the corpus, the word embedding space with additional corpus performs better on clustering. In contrast, the cluster is more distinctive. For the semantic direction Fig. 12, it is almost equivalent after fine-tuning, and it has no great influence on the semantic direction. Basically, all analogy pairs of words are close to the same position in the direction, and the lines are almost perpendicular.

In terms of semantic similar words in Fig. 11, the subtle influence of fine-tuning of the corpus can be seen in the *k*-nearest-neighbor words.

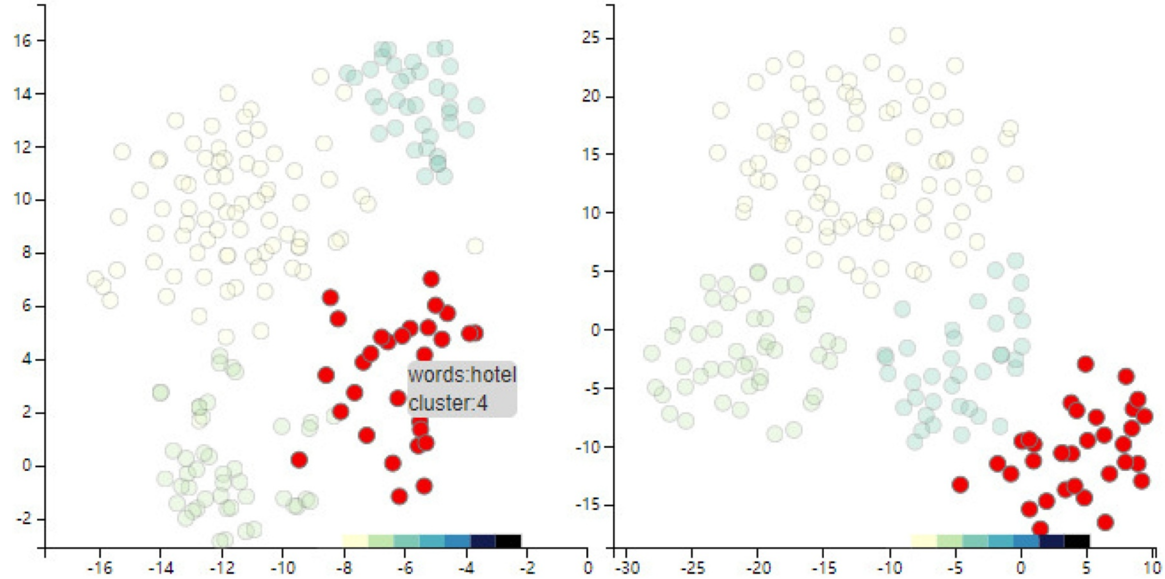

Fig. 10. The fine-tuning effect in the cluster views. The right adds the yelp comment corpus, which shows a better result in the cluster result.

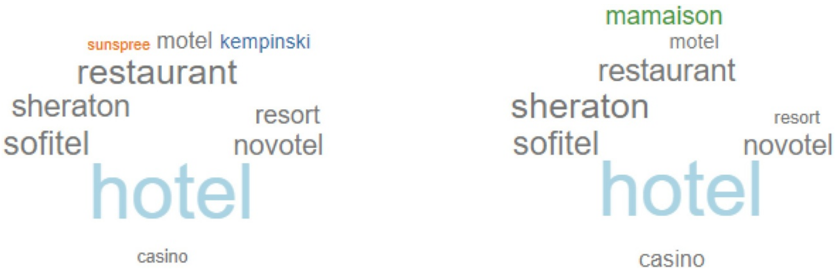

Fig. 11. Fine-tuning corpus changes in the semantic similar word view. Most semantically similar words are the same.

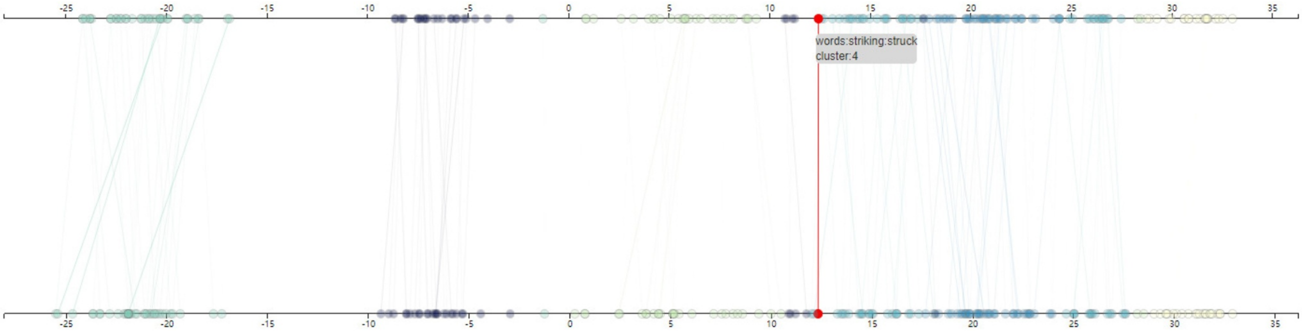

Fig. 12. The analogy semantic directions after fine-tuning. Two word embedding spaces are almost the same. The direction of the analogy pair *striking:struck*, in red, is perpendicular, and this means nearly no difference. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Due to the addition of comment data, the corpus repeatedly refers to the brand of the hotel so that the word of the hotel brand becomes closer in *hotel* semantic relations. The semantic similar words also can reflect the co-occurrence of some words, which mean that *sheraton* is closer in the space if the *sheraton hotel* is more likely used.

7. Conclusion and future work

In this paper, we designed an interactive visual analytic system, including the clustering view, semantic similar word view and analogy view, to visually understand, explore and compare word embeddings. We compare three different aspects of word embeddings, and find some interesting observations. In the cluster view, we find that the word *nanny* has the meaning of *babysitter* in most of the case rather than the labeled meaning *female goat*. The analogy view shows the analogy pair *striking:struck* is outlier and requires further exploration. What's more, we can see the latent semantic changes through the semantic similar word view. The example is that *hotel* changes from a functional place to the enjoyable location, which we may be not find in the dictionary. Still, our work has limitations. We focus on the semantic relations of word embeddings, rather the relation between spatial and semantic. Although we propose three views to present relations, we do not consider the non-linear relations between words.

For the future work, we will add some quantitative measurements for comparison, so that the users can understand more clearly. We also want to excavate more distributions of word embeddings. We plan to extend our scope of the system to fit more kinds of high-dimensional vector space and provide a more comprehensive visualization view of the space.

References

[1] G. E. Hinton, Learning distributed representations of concepts, in: Proceedings of the Eighth Annual Conference of the Cognitive Science Society, vol. 1, Amherst, MA, 1986, p. 12.

[2] W. Xu, A. I. Rudnicky, Can artificial neural networks learn language models?, in: Proceedings of the Sixth International Conference on Spoken Language Processing, ICSLP 2000, 2000.

[3] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155.

[4] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, In International Conference on Learning Representations, (2013).

[5] M. Kusner, Y. Sun, N. Kolkin, K. Weinberger, From word embeddings to document distances, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 957–966.

[6] J. Xu, Y. Tao, H. Lin, R. Zhu, Y. Yan, Exploring controversy via sentiment divergences of aspects in reviews, in: Proceedings of the IEEE Pacific Visualization Symposium (PacificVis), 2017, pp. 240–249. 10.1109/PACIFICVIS.2017.8031600.

[7] J. Xu, Y. Tao, Y. Yan, H. Lin, Vaut: a visual analytics system of spatiotemporal urban topics in reviews, J. Vis. 21 (3) (2018) 471–484.

[8] R. Socher, J. Bauer, C.D. Manning, et al., Parsing with compositional vector grammars, in: Proceedings of the Fifty-First Annual Meeting of the Association for Computational Linguistics, (Volume 1: Long Papers), 1, 2013, pp. 455–465.

[9] J. Pennington, R. Socher, C. Manning, Glove: global vectors for word representation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[10] S. Lai, K. Liu, S. He, J. Zhao, How to generate a good word embedding, IEEE Intell. Syst. 31 (6) (2016) 5–14.

[11] O. Levy, Y. Goldberg, I. Dagan, Improving distributional similarity with lessons learned from word embeddings, Trans. Assoc. Comput. Linguist. 3 (2015) 211–225.

[12] J. Mu, S. Bhat, P. Viswanath, All-but-the-top: simple and effective postprocessing for word representations. (2017) arXiv:1702.01417.

[13] A. Gittens, D. Achlioptas, M.W. Mahoney, Skip-gram-zipf+ uniform= vector additivity, in: Proceedings of the Fifty-Fifth Annual Meeting of the Association for Computational Linguistics, (Volume 1: Long Papers), 1, 2017, pp. 69–76.

[14] A. Globerson, N. Tishby, Sufficient dimensionality reduction, J. Mach. Learn. Res. 3 (2003) 1307–1331.

[15] X. Rong, E. Adar, Visual tools for debugging neural language models, in: Proceedings of the ICML Workshop on Visualization for Deep Learning, 2016.

[16] S. Arora, Y. Liang, T. Ma, A simple but tough-to-beat baseline for sentence embeddings, in: Proceedings of the International Conference on Learning Representations, 2017.

[17] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: Proceedings of the Twentieth ACM SIGKDD International Conference On Knowledge Discovery and Data Mining, ACM, 2014, pp. 701–710.

[18] S. Liu, X. Wang, C. Collins, W. Dou, F. Ouyang, M. El-Assady, L. Jiang, D. Keim, Bridging text visualization and mining: a task-driven survey, IEEE Trans. Vis. Comput. Gr. PP (99) (2018) 1-1.

[19] E. Alexander, M. Gleicher, Task-driven comparison of topic models, IEEE Trans. Vis. Comput. Gr. 22 (1) (2016) 320–329.

[20] A. Smith, T. Hawes, M. Myers, Hiearchie: visualization for hierarchical topic models, in: Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces, 2014, pp. 71–78.

[21] Y. Yang, Q. Yao, H. Qu, Vistopic: a visual analytics system for making sense of large document collections using hierarchical topic modeling, Vis. Inform. 1 (1) (2017) 40–47.

[22] C. Collins, F.B. Viegas, M. Wattenberg, Parallel tag clouds to explore and analyze faceted text corpora, in: Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, IEEE, 2009, pp. 91–98.

[23] E. Alexander, J. Kohlmann, R. Valenza, M. Witmore, M. Gleicher, Serendip: topic model-driven visual exploration of text corpora, in: Proceedings of the IEEE Conference on Visual Analytics Science and Technology, IEEE, 2014, pp. 173–182.

[24] A. Smith, J. Chuang, Y. Hu, J. Boyd-Graber, L. Findlater, Concurrent Visualization of Relationships between Words and Topics in Topic Models, in: Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces, 2014, pp. 79–82.

[25] X. Wang, S. Liu, J. Liu, J. Chen, J. Zhu, B. Guo, Topicpanorama: a full picture of relevant topics, IEEE Trans. Vis. Comput. Gr. 22 (12) (2016) 2508–2521.

[26] J. Choo, C. Lee, C.K. Reddy, H. Park, Utopian: user-driven topic modeling based on interactive nonnegative matrix factorization, IEEE Trans. Vis. Comput. Gr. 19 (12) (2013) 1992–2001.

[27] S. Liu, J. Yin, X. Wang, W. Cui, K. Cao, J. Pei, Online visual analytics of text streams, IEEE Trans. Vis. Comput. Gr. 22 (11) (2016) 2451–2466.

[28] J. Stasko, C. Görg, Z. Liu, Jigsaw: supporting investigative analysis through interactive visualization, Inf. Vis. 7 (2) (2008) 118–132.

[29] F. Heimerl, M. John, Q. Han, S. Koch, T. Ertl, Docucompass: effective exploration of document landscapes, in: Proceedings of the IEEE Conference on Visual Analytics Science and Technology, IEEE, 2016, pp. 11–20.

[30] C. Collins, S. Carpendale, G. Penn, Docuburst: visualizing document content using language structure, Computer Graphics Forum, 28 Wiley Online Library, 2009, pp. 1039–1046.

[31] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F.B. Viégas, M. Wattenberg, Embedding projector: Interactive visualization and interpretation of embeddings. (2016) arXiv:1611.05469.

[32] S. Liu, P.T. Bremer, J.J. Thiagarajan, V. Srikumar, B. Wang, Y. Livnat, V. Pascucci, Visual exploration of semantic relationships in neural word embeddings, IEEE Trans. Vis. Comput. Gr. 24 (1) (2018) 553–562.

[33] D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, N. Elmqvist, Conceptvector: text visual analytics via interactive lexicon building using word embedding, IEEE Trans. Vis. Comput. Gr. 24 (1) (2018) 361–370.

[34] W.L. Hamilton, J. Leskovec, D. Jurafsky, Diachronic word embeddings reveal statistical laws of semantic change, in: Proceedings of the Fifty-Fourth Annual Meeting of the Association for Computational Linguistics, (Volume 1: Long Papers), 1, 2016, pp. 1489–1501.

[35] L.v.d. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.

[36] C. Li, X. Dong, X. Yuan, Metro-wordle: an interactive visualization for urban text distributions based on wordle, Vis. Inform. 2 (1) (2018) 50–59.

[37] G.A. Miller, Wordnet: a lexical database for english, Commun. ACM 38 (11) (1995) 39–41.