KU LEUVEN

FACULTY OF SCIENCE

Comparison of dimensionality reduction methods on skip-gram processed medical dataset

Chunyang Zhang

Supervisor: Prof. Jan Aerts

Thesis presented in
fulfillment of the requirements
for the degree of Master of Science
in Statistics

Academic year 2015-2016

Preface

I am going to take this chance to express my sincere gratitude to everyone who has supported me throughout the making of my master thesis.

First of all, I would like to express my deepest gratitude to my promoter, Prof. Aerts, who gave me this excellent opportunity to do research on a topic of my interest. In addition, I also want to thank Prof. Aerts for providing me with insightful comments and for helping me to solve initial problems in the study. Without his help, I would not have been able to complete the study the way it is.

Secondly, I want to thank my daily supervisor, Thomas Moerman who was very patient with me and has provided me with vital feedback. On the weekly meetings, he raised important questions and gave me a lot of constructive suggestions. Every time I asked him questions, he always gave me fast feedback with lots of valuable details.

Moreover, I also want to thank my fiancé Matthias Busse who supported me throughout the past year and gave me the strength to pursue my goals.

Finally, I want to thank my friends and my family for their supports and the infinite love.

Summary

The medical system is entering into an era of 'Big Data'. In the medical sphere big data can refer to the information in medical and health science texts. There is a huge demand to mine information hidden in the texts (e.g., information of patients, patterns across patients, causal relationships of diseases) since this information may have significant values for drug development, disease diagnosis and healthcare management (ACCUMULATE, 2016). However, non-well-formed medical texts make it difficult to automatically recognise this crucial information, let alone the utility of the information. Therefore, people have started to look for advanced analytical tools to address this issue.

Recently, researchers have proposed to use natural language processing (NLP) on medical texts to extract valuable information from a myriad of datasets (Baud et al., 1992; Meystre et al., 2006; Miñarro-Giménez et al., 2015). One of the models in NLP, the so-called *distributed word representation*, has gained a lot of attention since it embeds words into a dense, real valued vector while capturing useful semantic and syntactic properties of the words and grouping similar words at the same time (Turian et al., 2010; Mikolov et al., 2013b). Although *distributed word representation* enables computers to derive meaning from human or natural language input, word vectors produced by this technique are always in high-dimensional spaces. Therefore, it is necessary to find an appropriate dimensionality reduction (DR) method to transform high-dimensional datasets into a meaningful low-dimensional map thus allowing people to visualise vector datasets for the naked eye.

Traditional dimensionality reduction techniques are linear techniques such as principal component analysis (PCA) or multidimensional scaling which aim at finding a linear subspace of lower dimensionality to represent a high-dimensional dataset (Ghodsi, 2006). Due to their linearity, these algorithms tend to preserve large pairwise distances. However, the local pairwise distances of high-dimensional data are more reliable than the large pairwise distances when the intrinsic dimension of high-dimensional data lies on a non-linear manifold (Van der Maaten and Hinton, 2008a). To address the limitation of linear DR methods, researchers come up with nonlinear DR methods such as t-student stochastic neighbour embedding (t-SNE) and Isomap which tend to keep data points that are similar in high dimensional data still close together in the low dimensional map (Van der Maaten et al., 2008b).

In this study, we introduce five DR methods: (1) principal component analysis, (2) t-student stochastic neighbour embedding, (3) Isomap, (4) Local linear embedding (LLE) and (5) Laplacian Eigenmaps (LE). We apply these methods as a visualisation tool to the 50-dimensional medical dataset generated by the skip-gram method and make a comparison of the performance across methods. Afterwards, we use the analogy tests to evaluate the quality of the word vectors of the unstructured medical corpus. Our study demonstrates that in terms of clustering structure and the stability of the technique under parameters and data variations, t-SNE exhibits the strongest performance on the skip-gram processed medical dataset compared to other four DR techniques. The results in the semantic and syntactic tests show that similar words are close to each other while they do not display syntactic and semantic similarity - word vectors go in a similar direction and sometimes even the length of the vectors are also similar.

Table of Contents

List of figures

List of tables

List of Abbreviations

CBOW	Continuous Bag-of-words Model
CCA	canonical correlations analysis
DR	Dimensionality Reduction
DBSCAN	Density-based spatial clustering of applications with noise
LE	Laplacian Eigenmaps
LLE	Local linear embedding
LDA	Fisher’s linear discriminant analysis
NLP	Natural Language Processing
MDS	multidimensional scaling
PCA	Principal component analysis
PCs	Principal components
t-SNE	T-student stochastic neighbour embedding
SNE	Stochastic neighbour embedding

1 Introduction

With the increase of unstructured medical texts in size and complexity, a large amount of crucial medical information (e.g., patterns across patients, causal relationships of diseases) is buried in these texts. Studies have found that the buried information may have specific applications in drug development, patient treatment and healthcare management (ACCUMULATE, 2016). Therefore, it is becoming extremely important to help companies, organizations and individuals to automatically recognize valuable information in a myriad of non-well-formed medical texts. Recently, researchers have proposed to use *natural language processing* (NLP) on medical files to recognize crucial patterns in the information hidden in the datasets (Baud et al., 1992; Meystre, 2006; Miñarro-Giménez et al., 2015). The goal of natural language processing is to design an appropriate algorithm to make computers understand natural language text or speech to perform a myriad of tasks (Chowdhury, 2003). At the core of any NLP task is how we represent the words as input to our models. Many natural language processing techniques treat words as discrete atomic units, which ignores the relationships that may exist between the individual words (Brants et al., 2007). Recently, with the progress of machine learning techniques, more complex NLP models are feasible, in which one of the most successful concepts is to use *distributed representations of words* (Mikolov et al., 2013a). The underlying idea of *distributed word representations* is to represent each word *w* in vocabulary *V* as a continuous-value vector of dimensionality *d* (*d* is smaller than *V*) (Qu et al., 2015).

Although *distributed word representation* enables computers to derive meaning from human or natural language input, word vectors produced by this technique are always on high-dimensional spaces. *The curse of dimensionality* reveals that the fast increase of dimensionality of data will make valuable data becoming 'sparse' (Bellman, 1961; Steinbach, 2003). This phenomenon has a negative effect on obtaining a reliable result based on the data. Dimensionality reduction (DR) methods have gained a lot of attention to address this issue since it can transform high-dimensional data into a meaningful reduced representation, thereby helping people to gain insights from the data. DR methods are divided into two types, linear and nonlinear methods. The nonlinear DR methods are always thought to have a stronger performance than linear methods and some research has confirmed that the nonlinear DR techniques perform well on nonlinear datasets (Brun et al., 2003; Lim et al., 2003; Niskanen and Silvén, 2003; Duraiswami and Raykar, 2005). However, it is not always the case. Some researchers find that nonlinear DR techniques failed in some datasets or they do not outperform the linear DR method 'PCA' on natural datasets (Lim et al., 2003; Van der Maaten, 2008b). Therefore, when it comes to a natural dataset, it is necessary to perform a comparison of dimensionality reduction methods to find an appropriate method.

In this study, we introduce five DR methods (PCA, t-SNE, Isomap, LE and LLE) to the skip-gram processed, unstructured medical corpus whose texts are derived from Medscape. Firstly, we perform these DR methods on a randomly-chosen-words dataset to have a basic knowledge of cluster structures. Secondly, to check the performance of the methods in the specific domain, we use these methods on the six 'word lists' filtered dataset and evaluate the arrangement of groups in the projections. In addition, we study the stability of these methods under the cost function parameter and data change by evaluating geodesic distance and cluster structures variations. At the end, we use the DR method with the best performance to evaluate the quality of word vectors by analogy tests.

The paper will be structured as follows: First we give a literature review on word representation and dimensionality reduction methods, secondly we introduce the methodology of dimensionality reduction methods, thirdly we present our result of dimensionality reduction methods and word vector evaluation, finally we draw conclusions on our results.

2 Literature review

2.1 Word representation

A word representation is a NLP technique that deals with words. There are three types of word representation:

1. Distributional representation methods. They map a word *w* to a context word vector C_w based on a co-occurrence matrix *F* $(W \times C)$ between the word *w* and its context words, where *W* is the vocabulary size, each row F_w is the initial representation of word *w*, and each column F_c is some context (Turian et al., 2010). One can project the matrix *F* into a lower-dimensional matrix *f* $(W \times d)$, with $d \ll C$, using some function *g*. For example, Dumais (Dumais et al., 1988) uses the DR technique singular value decomposition to compute the matrix *f.*
2. Cluster-based representation methods. They induce clusters of words by applying either soft or hard clustering algorithms[1](Qu et al., 2015). Some of them use the same matrix as distributional methods. For example, Pereira (1993) uses the co-occurrence matrix firstly and then transforms this matrix into a cluster. One of the well-known cluster-based methods is *Brown clustering* which uses a hierarchical clustering algorithm to maximize the mutual information of bigrams (Brown et al., 1992).
3. Distributed representation methods. Instead of training words as discrete units, distributed representation methods help mapping words into dense, low-dimensional, and real valued vectors which are called 'word embeddings' (Turian et al., 2010; Qu et al., 2015). Each dimension of a word vector represents an intrinsic characteristic of the word. A good distributed representation method should capture useful semantic and syntactic properties of the words and at the same time keep similar words close. Rumelhard, Hinton and Willians (1986) starts to use word representation 'idea'. This 'idea' has since been used in different fields such as automatic speech recognition, machine translation and mobile text entry and achieved considerable success.

Word2vec is one of the most popular word representation models based on distributed representation models. It has gained a lot of attention because of its computationally-efficient ability (Richard et al., 2016). High-quality vector representations of words are obtained without providing any example datasets to train the machine in advance which is usually required in supervised deep learning. Instead of using the co-occurrence matrix directly, *word2vec* (Mikolov et al., 2013b) predicts surrounding words of every word within a window size '*m'*. Two architectures of *word2vec* are skip-gram models and continuous bag-of-words models (CBOW) (Mikolov et al., 2013a). The skip-gram model's aim is to predict context-words from a center word. For example, assuming a given sentence "The cat jumped over the puddle" in which the center word is "jumped", skip-gram predicts the surrounding words "The", "cat", "over", "the", "puddle" from this center word (Richard et al., 2016). As opposed to the skip-gram model, the CBOW model predicts a center word given the surrounding context, that is, using the context-words "The", "cat", "over", "the", "puddle" predicts the target word "jumped". The inversion

[1] Soft clustering: clusters may overlap
Hard clustering: clusters do not overlap

between the CBOW and the skip-gram influences the choice of the two models based on the size of datasets. The CBOW model is more suitable for smaller datasets since it takes the surrounding context as one observation and simplifies a lot of distributional information, while the skip-gram model treats each context word as a new observation, which makes the model work better for larger datasets (TensorFlow, 2016).

Mikolov et al. (2013c) demonstrate that there exist linguistic regularities in vector-space word representations. In general, there are two methods to evaluate the quality of word vectors: intrinsic and extrinsic evaluations. Intrinsic evaluations use specific tasks such as relatedness and analogy to measure the quality of word vectors and thereby providing the inner working knowledge of the word embedding techniques. A popular choice of intrinsic evaluations is analogy tests**.** Analogical tests show that similar words after an embedding technique are not only clustered together, but also these words present linguistic regularities and patterns. This phenomenon has been shown in some studies (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov et al., 2013c; Lomonaco, 2015). Extrinsic evaluations apply word embeddings on the real task such as semantic role labeling or part-of speech tagging to evaluate the quality of word vectors (Schnabel, 2015). Compared to intrinsic evaluations, extrinsic evaluations are relatively slow in computation because of elaborate tests (Zhai et al., 2015; Schnabel, 2015). Since extrinsic tasks are relatively time-consuming and difficulty to work well, we only choose intrinsic evaluations in our study.

2.2 Dimensionality reduction methods

One essential component of the data analysis is to get an intuition on how those data is arranged in the data space via visualisation techniques, which improves the human comprehension of data. Real-world data is usually high dimensional. Having too many variables can sometimes be described as a *curse of dimensionality* (Bellman, 1961). Therefore, finding an appropriate visualisation tool is important for the visualisation of high-dimensional data in different fields of research. Over the last few decades, a lot of techniques have been proposed. For instance, iconographic techniques such as *Chernoff faces* and pixel-based techniques help to decrease dimensions of data. However, the reduced dimensionality generated by these techniques are more than two dimensions, which is still difficult for people to interpret the results (Chernoff, 1973; Keim, 2000). A popular way to perform a visualisation of high-dimensional data are DR methods. DR methods transform high-dimensional data into lower-dimensional representations which preserves as much information as in the original data (Fodor, 2002; Ghodsi, 2006). DR techniques are divided into linear techniques and nonlinear DR methods.

2.2.1. Linear dimensionality reduction

Linear DR methods map linear representations of high-dimensional data to a lower-dimensional space. Assuming *n d*-dimensional observations, each observation is represented by **x**=(x_1, …, x_p)^T and we define *i* and *j* as the number of dimensions and the number of observations, respectively (Fodor, 2002)[2]:

The mean of the *i*th random variable: $\widehat{\mu} = \overline{x_i} = 1/n\sum_{j=1}^{n} x_{i,j}$,

The standard deviation of the *i*th random variable: $\widehat{\sigma_i} = 1/n\sum_{j=1}^{n}(x_{i,j} - \overline{x_i})^2$,

Standardize the observation $x_{i,j}$: $(x_{i,j} - \widehat{u_i})/\widehat{\sigma_i}$.

For linear techniques, we will get a *n**k* ($k \leq p$) data *y* which is a linear combination of the

[2] The equations in this page and next page are taken from Fodor, 2002 (p1-2).

original variables

$$y_i = a_{i1}x_1 + ... + a_{ip}x_p \text{ , for } i\text{=1, …, } k$$

Simplifying the equation to $y_i = ax$, where a_{k*p} is a linear transformation weight matrix,

In terms of an *n*p* observation matrix X, we have

$$y_{ij} = a_{i1}x_{1j} + ... + a_{ip}x_{pj} \text{ , for } i\text{=1, …, } k\text{, and j=1,…,n}$$

Equivalently, $y_{k\times n} = a_{k\times p}x_{p\times n}$

Linear DR methods have been developed in domains such as biology, chemistry, landscape ecology and artificial intelligence research for over a century (Riitters et al, 1995; Du et al., 2006; Kumar et al., 2014). Du et al. (2006) proposes amino acid principal component analysis in protein structure classification. This method maps high-dimensional amino acid composition data into an orthogonal lower-dimensional space (Du et al., 2006). Factor analysis (Riitters et al, 1995) is used to identify the common factors of landscape pattern and structure by reducing fifty-five dimensional landscape data. Linear DR methods are popular partly due to their simple geometric interpretation of high-dimensional data in a low-dimensional space (Cunningham and Ghahramani, 2015).

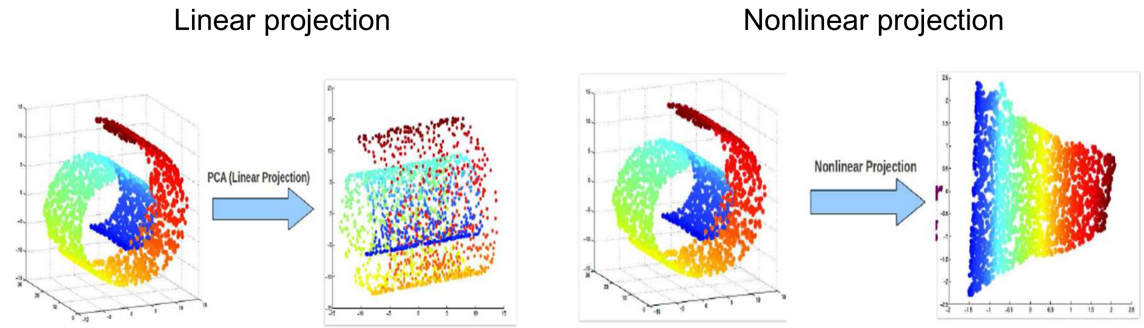
A lot of linear dimensionality reduction methods have been developed such as the PCA, factor analysis, multidimensional scaling (MDS), Fisher's linear discriminant analysis (LDA), canonical correlations analysis (CCA), and others.

PCA is by far one of the most popular linear DR methods plays an important role in data analysis. The goal of PCA is to find a new set of variables, the *principal components* (PCs), that are orthogonal and linear combinations of the original dimensions. PCs are computed by using a general eigendecomposition of a covariance or a correlation matrix. A decision on using a covariance or a correlation matrix in PCA influences the result of low-dimensional representations (Borgognone et al., 2001). Notably PCA is sensitive to the scale difference of variables. Assuming a set of variables in a dataset with widely varying scales (e.g. length, temperature, blood pressure), the choice of units of variables will decide the structure of the principal components derived from the covariance matrix (Brian and Torsten, 2011). Moreover, the first principal component will mainly explain the variables with the largest variance. One can only use covariance matrix when the variables are on the similar scale. However, it is not always the case in practice. Instead, in practice, one should derive principal components from correlation matrix, R, that rescale the variance of the original data to a unit. However, we still use the covariance matrix although the scales of variables are different when the variance of the original variables is important (Brian and Torsten, 2011).

2.2.2. Nonlinear dimensionality reduction

In terms of which distances to preserve, linear DR methods such as PCA tend to retain large pairwise distances, which means that data that are not similar in the original data are still dissimilar in a lower subspace (Cunningham and Ghahramani, 2015; Ghodsi, 2006). Although linear DR methods are able to extract effective features of high-dimensional data and display these features in a low-dimensional map, these large distances are actually not informative when the data is complex (Van der Maaten and Hinton, 2008a). For example, when the high-dimensional data lies on a nonlinear manifold, their Euclidean distances in the high dimensional space may not accurately reflect their intrinsic similarities (Tenenbaum et al., 2000). It is usually more important to retain local distances that are used to make sure similar data points in high dimensional data still close together in the low-dimensional map (Rai, 2011; Ghodsi, 2006). This is typically not possible with a linear DR method. It is proven by the Swissroll dataset in figure 1. It shows that the linear projection PCA cannot capture intrinsic structures while in nonlinear projection clusters are presented well. Nonlinear DR methods have gained more attention since these kinds of datasets are very common in machine learning.

Figure 1: PCA and nonlinear technique in a nonlinear manifold



Linear projection

Nonlinear projection

Source: Rai, 2011: p13-14.

Based on the relation to linear methods, nonlinear DR methods can be defined as: 1) nonlinear DR methods developed from linear DR methods, such as kernel PCA; 2) manifold based methods such as Isomap, LLE and t-SNE.

2.2.3 Challenges and current solutions

Due to linearity of linear DR methods, their algorithms tend to preserve large pairwise distances. However, when the intrinsic dimension of high-dimensional data lies on a non-linear manifold, the large pairwise distances that DR methods tend to preserve are not reliable (Van der Maaten and Hinton, 2008a). To address this issue, researchers have come up with a variety of nonlinear DR methods such as t-SNE (Van der Maaten and Hinton, 2008a), Isomap (Tenenbaum et al., 2000) and Local linear embedding (Roweis and Saul, 2000). The main drawback of PCA is that the size of the covariance matrix is proportional to the dimensionality of the data points (Mishra et al., 2012). As a result, in a dataset with a large number of data points or very high dimensions, it becomes impossible or very slow to compute the eigenvectors. In the situation when the number of data points is smaller than the number of dimensions, researchers switch from PCA to classical scaling since classical scale deals with the number of data points instead of the number of dimensions (Torgerson, 1952). When the dataset is with high dimensions, an iterative method 'simple PCA' is be used as a fast approximation for PCA (Partridge and Calvo, 1997).

Although the nonlinear DR method 't-SNE' is more competitive than PCA in many aspects (e.g. clustering structure and stability), its computational requirement and memory are more complex than PCA when the number of data points is large (Van der Maaten and Hinton, 2008a). In the choice of DR methods, it is also important to consider how time consuming each method is. To address this limitation of t-SNE, Van de Maaten (2013) comes up with Barnes-Hut-SNE by using a sparse distribution[3] to approximate the pairwise similarity probability p_{ij}[4]. Since a Gaussian distribution is used in the computation of the pairwise probability p_{ij}, p_{ij} will

[3] A sparse distribution in which infinitesimal pairwise similarity probabilities are changed to zero (Van de Maaten, 2013).

[4] A pairwise similarity probability is used to measure the similarity between two data points. In a high-dimensional space where there are many objects, we take a high-dimensional object called x_i and center a Gaussian at x_i. Next the probabilities of all the other points x_j under this Gaussian are computed by dividing a density of by x_i and x_j the sum of these densities. (Van der Maaten, L. and G. Hinton, 2008a)

be almost infinitesimal if two points are widely separated (Van der Maaten and Hinton, 2008a). Therefore, replacing pij by a sparse approximation will not have considerable influence on the quality of the result embeddings (Van de Maaten, 2013). Another drawback of t-SNE is non-parametric property. It causes an out-of-sample extension problem: when the old dataset is updated with some new data points, we cannot map new high-dimensional data points into the existing low-dimensional space directly (Strange and Zwiggelaar, 2011). This problem can be solved by applying parametric t-SNE (Van der Maaten, 2009). Isomap suffers from:

1) topological instability which will influence the construction of the neighbourhood graph G (Van der Maaten et al., 2008b). To solve this problem, Saxena et al. (2004) proposes a new algorithm which uses local linearity property of the manifolds and only keeps nearest neighbours that meet local linearity assumption of neighbourhood graph G.

2) "holes" in the manifold which impair the performance of Isomap. This weakness can be overcome by tearing or cutting the 'circular' manifolds (Lee and Verleysen, 2005).

Both linear and nonlinear DR methods have their advantages and disadvantages. When it comes to a natural dataset, it is necessary to perform a comparison of dimensionality reduction methods to find an appropriate method.

3 Methodology

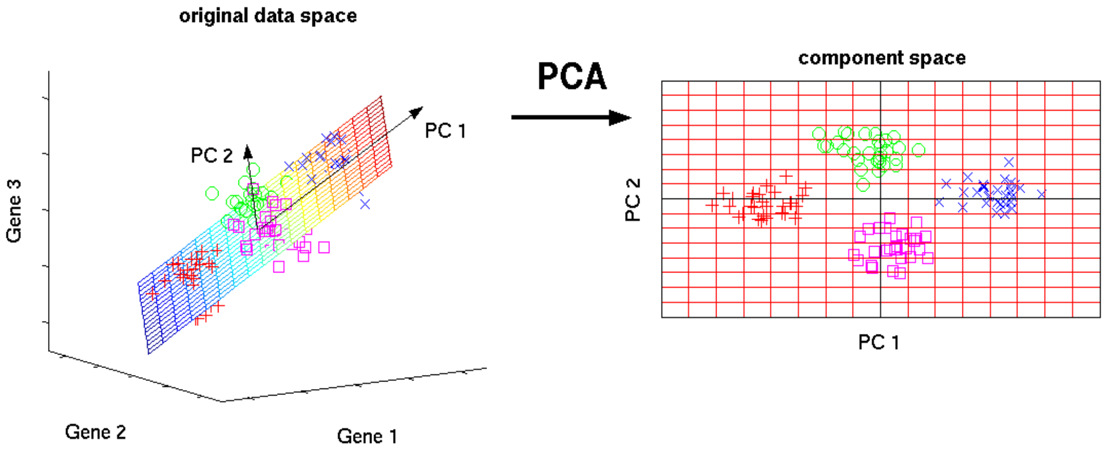
We introduce one linear DR methods and four nonlinear DR methods in 3.1 and 3.2. Then we present the analogy tests that are used to evaluate the quality of word vector in 3.3.

3.1 Linear dimensionality reduction methods

3.1.1 Principal component analysis (PCA)

The basic aim of PCA is to map a high-dimensional dataset where a set of variables are correlated into a low-dimensional subspace where a new set of variables are uncorrelated while explaining the original variation as much as possible in the new subspace (Brian and Torsten, 2011; Wolfgang Karl and Simar. 2012). This aim can also be described as finding a new set of variables, the *principal components* (PCs), that are orthogonal and linear combinations of the original dimensions. A graphical representation of a PCA transformation is shown in figure 2.

Figure 2: Graphical representation of a PCA transformation



Source: Scholz, 2006: p1.

The PCs is arranged in a sorted order according to the variance they explain. The first PC in figure 2 is the linear combination of the original variables whose sample variance is largest among all PCs. The second PC is also the linear translation of the original variables that accounts for a maximal proportion of the remaining variance and is orthogonal to the first PC. The rest PCs are described in the similar way. The PCs are computed by performing a general *eigendecomposition* of the covariance matrix of the original data.

The number of PCs is as same as the number of the original variables. The general hope of PCA is that the first few PCs will explain a substantial variance in the original data and can hence be used to form new coordinates (Brian and Torsten, 2011). Although we lose some information by throwing away some PCs, we will not lose too much if the eigenvalues of these PCs are small. We can select the appropriate number of PCs to maintain a given percentage of the total variation explained according to *scree* diagram plots, i.e. the cumulative proportions plots. The selection of the number of PCs can also be determined by fixing a threshold λ_0, only keeping the principal components whose eigenvalues are larger than λ_0 (Fodor, 2002).

3.2 Nonlinear dimensionality reduction methods

3.2.1. T-distribution stochastic neighbor embedding (t-SNE)

T-SNE tends to preserve small pairwise distances, instead of focusing on preserving large pairwise distances (Van der Maaten and Hinton, 2008a). Basically we are trying to make sure that the nearest neighbours of a point in the original data are also nearest neighbours of the point in a low-dimensional map. T-SNE is a method that develops from stochastic neighbor embedding (SNE).

Stochastic neighbor embedding (SNE) is first proposed by Hinton and Salakhutdinov (2002). The mechanism is as follows: In a high-dimensional space where there are many objects, we take a high-dimensional object called x_i and center a Gaussian at x_i. Next the probabilities of all the other points x_j under this Gaussian are computed by dividing a density by the sum of these densities. The variance σ_i is selected by a binary search where the user specifies a fixed perplexity and produces P_i accordingly (Van der Maaten and Hinton, 2008a). The perplexity can be interpreted as the effective number of neighbours of one point and the value is often set between 5 and 50 (Van der Maaten and Hinton, 2008a). The probability distribution of pairs of points *i, j* is given as[5]

$$P_{j/i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}.$$

We set the value of $P_{i/i}$ equal to zero as we only deal with pairwise similarities. If two data points are relatively close, the probability $P_{j/i}$ is high, while the probability will be almost infinitesimal if two points are widely separated.

In the low dimensional space, the dots that represent points in a high-dimensional space are written as y_i and y_j. We set the variance of the Gaussian in the conditional probability $q_{j/i}$ to ½ and the rest of operation is the same as that in the high-dimensional space. The conditional probability $q_{j/i}$ is given as

$$q_{j/i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)}.$$

We also set the value of $q_{i/i}$ equal to zero.

Ideally we want to match the conditional probabilities $p_{j/i}$ in the low-dimensional map with $q_{j/i}$. The mismatch between two probabilities is measured by the *Kullback-Leibler*:

$$C = \sum_i KL(\mathrm{P}_i \| Q_i) = \sum_i \sum_j p_{j/i} \log \frac{p_{j/i}}{q_{j/i}}.$$

where P_i and Q_i are the conditional probabilities over all data points in high-dimensional and low-dimensional spaces, respectively.

If the two distributions are the same, the cost function is zero. SNE minimizes the cost function by the gradient descent method. Due to the asymmetry of *Kullback-Leibler* divergence, different types of projection error in pairwise similarities are weighted differently (Van der Maaten and Hinton, 2008a). If the value of $p_{j/i}$ between two objects in the original data is large, it is better to make sure that $q_{j/i}$ is not zero. Otherwise the *Kullback-Leibler* divergence will be large and we need to pay a very large penalty for having two objects that are similar in the

[5] The equations in this page and next page are taken from Van der Maaten and Hinton, 2008a (p2581-2584).

original data, but far apart in the map. However, it does not work the other way around. If two points are dissimilar, we do not pay a cost for putting them closer together since $p_{j/i}$ outside is infinitesimal. Hence, SNE mainly preserves local similarity structure of the data.

Although SNE displays a nice visualisation, the optimisation problem of the cost function and the crowding problem restrict its application in more fields. The cost function of t-SNE is different from SNE in two ways (Van der Maaten and Hinton, 2008a): 1) it changes the cost function of SNE into a symmetrised version, which makes the optimization of the cost function easier. 2) it applies a t-distribution instead of a Gaussian in the low-dimensional map to compute the similarity probabilities.

The symmetric SNE

Instead of using the conditional probabilities $p_{j/i}$ and $q_{j/i}$, the symmetric SNE uses the joint probabilities p_{ij} in the high-dimensional space and the joint probability q_{ij} in the low-dimensional space,

$$C = \sum_i KL(\mathrm{P}\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}},$$

$$p_{ij} = \frac{p_{j/i} + p_{i/j}}{2n}.$$

This type of SNE is called symmetric because of its properties $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji}$. The gradient of the symmetric SNE is simpler than SNE

$$\frac{\delta C}{\delta y_i} = 4\sum_j (\mathrm{p}_{ij} - \mathrm{q}_{ij})(\mathrm{y}_i - y_j) \text{ in symmetric SNE,}$$

$$\frac{\delta C}{\delta y_i} = 2\sum_j (\mathrm{p}_{j/i} - q_{j/i} + p_{i/j} - q_{i/j})(\mathrm{y}_i - y_j) \text{ in SNE.}$$

The crowding problem

The crowding problem occurs when data is intrinsically high-dimensional but we try to model the local structure of this data in a low-dimensional map. For instance, we try to map high-dimensional data whose manifold has ten intrinsic dimensions into a two-dimensional map. Dissimilar points in the original data have to be model too far apart in the map. In order to preserve as much as information in the original data, an attraction force is produced to get these too far away map points a bit closer together (Van der Maaten and Hinton, 2008a). Although these attractive forces are small, if there are thousands of points that are all trying to be closer, it will cause crowding problem in the centre.

In t-SNE, it uses a student t-distribution with one degree of freedom, a heavy-tailed distribution, to solve this problem,

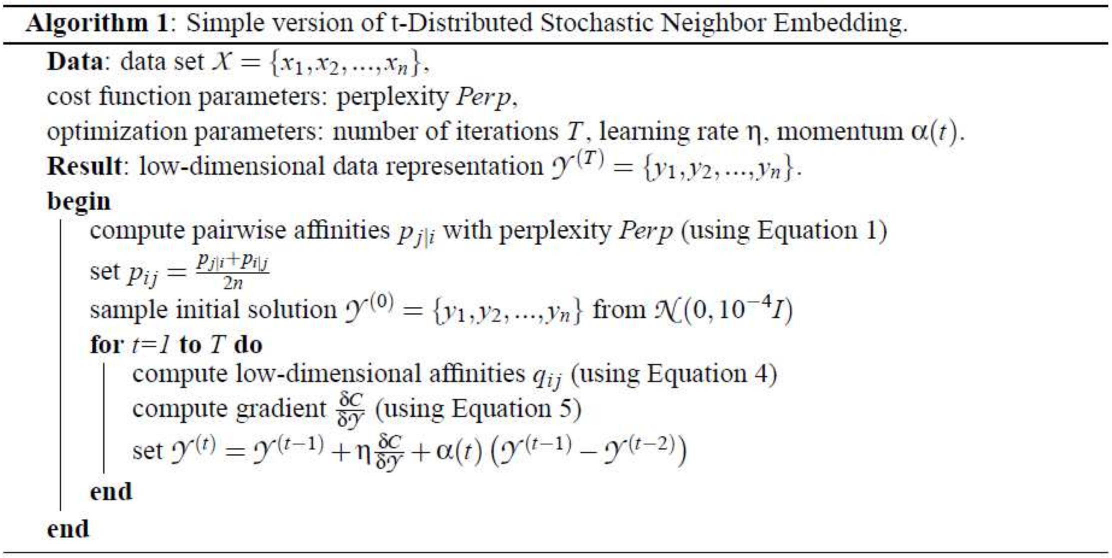
$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}.$$

Because of the property of t-distribution, dissimilar objects in the original data are allowed to be modeled by a much larger distance in the map than in SNE. For example, the distance between two points in the data is 20 and the corresponding similarity probability is 0.1 and then to get the same density of 0.1, the distance is larger in the map.

Whether DR techniques are efficient and robust will have a strong effect on the resulting embedding. The more stable the embedding of DR methods under variations are, the easier it is to obtain reliable visual comprehension in datasets (Carcía-Fernández et al, 2013a). Therefore, we study the stability of dimensionality reduction under variations in parameters and data. This experiment is used as another evaluation of the performance of the DR methods. To find which parameters will have an effect on the robustness of t-SNE, we will learn from the algorithm of t-SNE.

The cost function in t-SNE is optimised by a simple gradient descent procedure. This procedure can be sped up by introducing a *momentum term* $a(\mathrm{t})$ and a *learning rate* η. The algorithm is shown in figure 3.

Figure 3: Simple version of optimization of the cost function in t-SNE



```
Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.
  Data: data set $X = \{x_1, x_2, \ldots, x_n\}$,
  cost function parameters: perplexity $Perp$,
  optimization parameters: number of iterations $T$, learning rate $\eta$, momentum $\alpha(t)$.
  Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$.
  begin
      compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
      set $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2n}$
      sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
      for $t=1$ to $T$ do
          compute low-dimensional affinities $q_{ij}$ (using Equation 4)
          compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
          set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta\frac{\delta C}{\delta \mathcal{Y}} + \alpha(t)\left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
      end
  end
```

Source: Van der Maaten and Hinton, 2008a: p2587.

In addition, the visualisation results can be improved by two means (Van der Maaten and Hinton, 2008a): 1) by early compression, which is to put the map points closer together at the beginning of optimization. When the distances between map points are small, clusters can go through other clusters easier in the iteration computation and hence it is good for exploring the structure of possible global organisations of the data. 2) by early 'exaggeration'. In order to model the relatively large corresponding p_{ij}, almost all of q_{ij} become larger. In consequence, clusters themselves in the map are formed closer and separate from other clusters, which creates more empty space in the map for clusters to more around in the iterations to find a better global visualisation. This problem can be solved by multiplying all of the p_{ij} by, for example the value 4, for the initial iterations.

Figure 3 shows that each of parameters perplexity *p*, the number of iterations, the momentum term $a(\mathrm{t})$ and the learning rate η can have an influence on the resulting embedding of t-SNE. In this study, we focus on studying the impact of perplexity and set the number of iterations, the momentum term $a(\mathrm{t})$ and the learning rate η to a fixed number.

3.2.2 Isomap

Isomap is a DR method that is similar to multidimensional scaling analysis (MDS) in that it preserves pairwise distances between data points. However, instead of using Euclidean distance in MDS, Isomap applies curvilinear distance, i.e. the distance between data points over the manifold. In Isomap, researchers first construct a neighbourhood graph G over all data points by connecting each data point X_i with its *k* nearest neighbours, where the value of *k* is defined by the users (Tenenbaum et al., 2000). Thereafter, the estimates of the curvilinear distances are computed as the shortest paths between all pairs of data points in the graph (Tenenbaum et al., 2000). *D*-dimensional representations are constructed by performing the eigendecomposition of the pairwise curvilinear distance matrix.

3.2.3 Local linear embedding (LLE)

Similar to Isomap, LLE also constructs a graph G that represents the data points in a high-dimensional dataset. However, compared to Isomap, LLE only focuses on local properties of the data. LLE assumes that the local properties of the manifold around data points fit a hyperplane in such a way each data point x_i can be reconstructed as a linear combination w_i (reconstruction weights) of its *k* nearest neighbours (Saul, 2001). Reconstruction weights are stable to translation, rotation and rescaling (Saul and Roweis, 2003). Because of that, the reconstruction weights that are found in the high-dimensional data also work in reconstructing y_i from its neighbours in the low-dimensional space if the low-dimensional space preserves the local properties of the manifold (Van de Maaten et al., 2008b). Therefore, in order to find the low-dimensional data points y_i that represent high-dimensional data points x_i, we need to minimise the cost function

$$\phi(\mathrm{Y}) = \sum_i \left\| y_i - \sum_{j=1}^{k} w_{ij} y_{i_j} \right\|^2 \text{ subject to } \left\| y^{(\mathrm{k})} \right\|^2 = 1 \text{ for } \forall k$$ [6].

Roweis and Saul (2000) use the smallest *d* nonzero eigenvalues and their corresponding eigenvectors of the matrix $(I-W)^T(I-W)$ to obtain the *d*-dimensional data representation y_i, where *W* consists of reconstruction weights and *I* is n × n identity matrix.

3.2.4 Laplacian Eigenmaps (LE)

LE is a non-linear technique that preserves local properties of the manifold in the low-dimensional space. Similar to Isomap, LE also constructs an adjacency graph G that connects each data point x_i with its *k* nearest neighbours in the high-dimensional space. The aim of LE is to find a low-dimensional space in which the *k* nearest neighbours of a data point are the same as in the original data, which is achieved by adding weights to the cost function (Belkin and Niyogi, 2002; Wang, 2011). If data points x_i and x_j in the graph are connected, the Gaussian kernel function is used to compute the weight of these two points (Belkin and Niyogi, 2002),

$$w_{ij} = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}.$$

A sparse matrix *W* is made up of weights between all these two data points.

The cost function below is minimised to find the low-dimensional data representations y_i,

$$\phi(\mathrm{Y}) = \sum_{ij} \left\| y_i - y_j \right\|^2 w_{ij}$$ [7].

The data points that are close in the original space will obtain a high weight. Thus the corresponding low-dimensional data representations will have a higher contribution in the cost function.

Assuming that the degree matrix *M* of *W* is a diagonal matrix and the entries of *M* are equal to

[6] The equation is taken from Van de Maaten et al., 2008b (p8).

[7] The equation is taken from Van de Maaten et al., 2008b (p9).

the sum of rows of *W*, $m_{ii} = \sum_j w_{ij}$, and the graph Laplacian *L* of *W* is equal to M-W (Van de Maaten et al., 2008b), the cost function can be written as

$$\phi(\mathrm{Y}) = \sum_{ij} \left\| y_i - y_j \right\|^2 w_{ij} = 2Y^T L Y \text{ subject to } Y^T MY = I_n$$[8].

Hence, the question of finding the *d*-dimensional data representations is converted to look for the smallest *d* nonzero eigenvalues and corresponding eigenvectors of the graph Laplacian *L*.

3.3 DBSCAN

DBSCAN is a clustering algorithm that can detect clusters of arbitrary shape based on the density of points and handle the noise points effectively (Ester et al., 1996). The performance of DBSCAN is sensitive to the choice of the two parameters *eps*, which determines how far to search for neighbour points given a point, and *MinPts*, which defines a minimum number of points that should present in the neighbourhood of a given point to form a cluster (Karami and Johansson, 2014). Ester et al. (1996) proposed to use the sorted *k*-dist graph in which the *k*-nearest neighbour distances of each point are plotted in an ascending order to estimate the optimal value *eps* after fixing *MinPts* to a certain number. The optimal *eps* corresponds to a sharp change in the sorted *k*-dist graph.

3.4 Intrinsic evaluation of word vectors

Analogy tests measure syntactic and semantic relationships by simple algebraic operations on the word vectors (Mikolov et al., 2013a; Mikolov et al., 2013b). Figure 4 illustrates an example of analogy tests - countries-capitals test. A corpus of countries and capitals is converted into real valued vectors and these vectors are projected into a two-dimensional map, which is shown in figure 4. In Figure 4, we observe that neighbouring countries (Germany + France and Spain + Portugal) are closer than other countries and the semantic regularities between a country and its corresponding capital city show linear patterns, e.g. Pairs is to France as Madrid is to Spain. Mikolov et al. (2013c) propose a simple *vector offset method* to explain the algorithm of this vector operation. Given an analogy question with an unknown term *d*, *a:b::c:d* that *d* is similar to *c* in the same sense as *b* is similar to *a.* The best answer for the unknown term *d* is the word whose word representation maximizes the cosine similarity:

$$d = \arg\max_i = \frac{(x_b - x_a + x_c)^T x_i}{\left\| x_b - x_a + x_c \right\| \left\| x_i \right\|}$$[9].

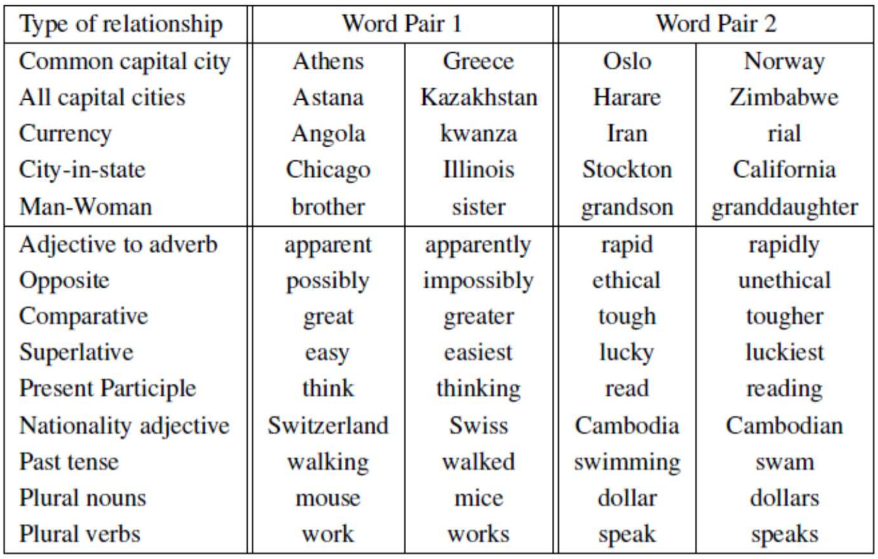
According to the vector offset method, the countries and capitals example can be explained in such a way that the vector "*Madrid*" *vec("Madrid")* is closer to the result of the vector computation *vec("Pairs")-vec("France")+vec("Spain")* than any other word vectors based on the cosine distance.

There are two categories in analogies: syntactic and semantic regularities. Some examples of syntactic and semantic questions are shown in table 1.

[8] The equation is taken from Van de Maaten et al., 2008b (p9).

[9] The equation is taken from Mikolov et al., 2013c (p748).

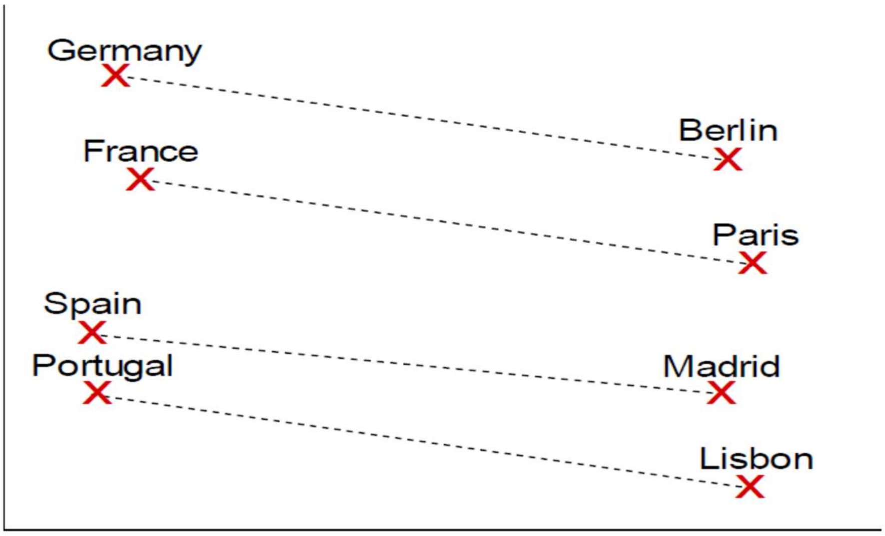
Table 1: Syntactic and semantic relationship test set



Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Source: Mikolov et al, 2013a: p6.

Figure 4: The regularities of vector space representations in a two-dimensional map



Source: Miñarro-Giménez et al., 2015: p3.

4 Results

In this chapter, we will first present the performance of the five DR methods on the 5000 randomly-chosen-words dataset and the six 'word lists' filtered dataset in 5.1.2 and 5.1.3. Then, the study of the stability of methods is shown in 5.2. At the end, the results of analogy tests are presented in 5.3

4.1 Dimensionality reduction methods

To evaluate the performance of DR methods on the *word2vec* processed medical corpus, we have selected five DR methods in which there is one linear technique, PCA, and four nonlinear techniques, t-SNE, Isomap, LLE and LE.
The settings of the cost function parameters we use in the study are listed in table 2.

Table 2: Cost function parameter settings in the study

Technique	Parameters	settings
PCA	None	None
t-SNE	Perplexity (P): the effective number of neighbours	30
Isomap LE LLE	k: number of nearest neighbours	12

Source: the table is created based on the table in Carcía-Fernández (2013), p97.

4.1.1 Description of the data

The medical dataset includes 147,764 unstructured medical terms derived from Medscape, a website that contains web text crawled from Medscape articles. Machine learning researchers applied the word2vec continuous skip-gram model to train this dataset with seven window sizes, 1, 2, 4, 8, 16, 32, 64 and thus obtained seven datasets with a different window size where the meaning of each term is represented by a 50-dimension vector. In the 5000 randomly-chosen-words dataset or six 'word lists' filtered dataset, class information is added to each term. However, the class information is not applied in the algorithm of DR methods, but only used in the colour selection for data points in projections, which provides us with a way to obtain information on how similar data points are arranged in the map.

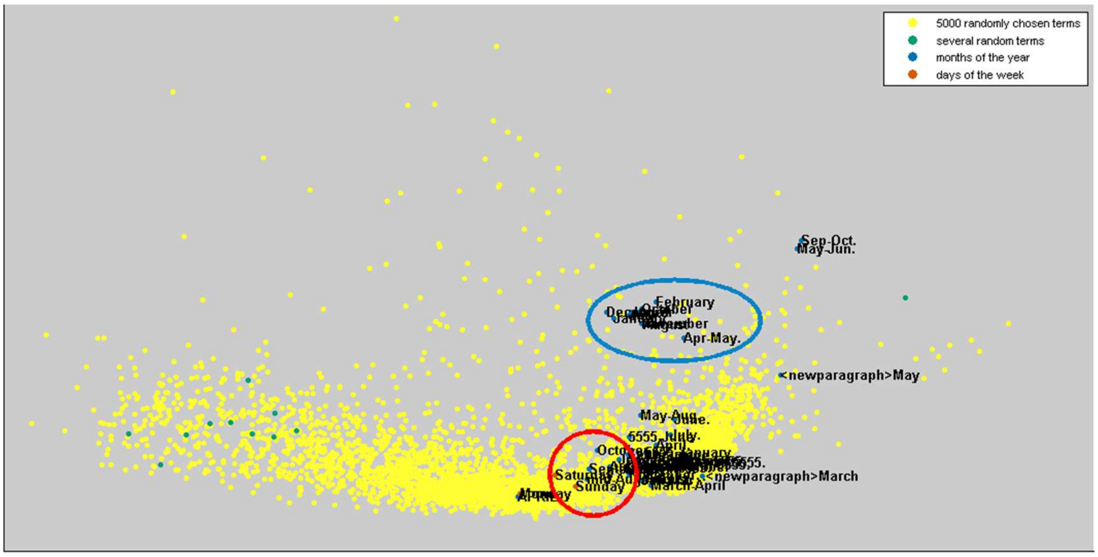
4.1.2 A 5000 randomly-chosen-words dataset

There are 147,764 terms in our dataset. Obviously, if we visualize all terms at the same time, a lot of points in the plot will be overlapping and thus the plot will be hard to read. To make the plot easy to visually analyse, we firstly picked three categories of word lists (1. days of week; 2. months of year; 3. several random English terms). Afterwards, we form a three-category target dataset by selecting the rows from the word embedding dataset whose names of the entries correspond to the words in the word lists. We choose 5000 terms randomly from the original dataset to form a 5000-term dataset in which we delete words from our word lists. At the end, we combine the three-category target dataset with 5000-term dataset which we will use in this study. Within the group that consists of months of year, there are two subgroups, one is words themselves like *February* and another is words with some 'additions' (prefix and suffix) like *Sep-Oct*.

Because of the limited space, we only present the results of five DR methods on the dataset with window size 1 in Figure 5-8 and the rest of the results from the other six different window-size datasets are presented in the appendix A (figure 29-32). The performance of the five DR methods on the other six different window-size datasets are similar to that in the dataset with window size 1. Figure 5 and 7 show that the difference between the performance of PCA and Isomap is small since the most data points in the 2-dimensional maps are distributed along the *x* axis, but Isomap projection diverges on the left of the distribution. Moreover, the month of the year data points without 'additions' in PCA and Isomap graph (highlighted with a blue circle) tend to stay close and separate from other months of the year terms with 'additions'. The data points that represent days of the week in the red circle stay close to the months of the year with 'additions'. The t-SNE generates a number of clear clusters, as seen in figure 6. As opposed to PCA where days of the week are mixed with the months of the year with 'additions', the t-SNE plot separates days of the week from other points with text, even though the term Friday is not close to the terms from the same group. Compared to three other DR methods, the performance of LE and LLE is unsatisfactory since LE arranges the data points from the same groups in a line instead of clustering the data points and LE just projects most elements of the dataset in a single line segment. The explanation is shown in conclusion part.

The results of t-SNE on seven datasets with a different window size in figure 9 reveal that data points that represent months of the year without 'additions' are always clustered together under the variation in window size. Moreover, the value of window size has an influence on the clustering information of the terms from days of the week. For instance, in window-size 1 plot, three terms, Monday, Sunday and Saturday cluster together and these are away from the term Friday while in window-size 8 plot, these four terms are scattered.

Figure 5: PCA on the dataset with window size 1



Note: We used four different colours to label 5000 randomly chosen terms, several random English terms, months of the year and days of the week. Yellow, green, blue and red dots represent the 5000 randomly chosen terms, several random terms, months of the year and days of the week, respectively. After we put text on the data points belonging to several random terms, months of the year and days of the week, we found that there exist some clustering structures. Therefore, we used blue and red circles to mark the position of months of the year without 'additions' and days of the week.

Figure 6: T-SNE on the dataset with window size 1

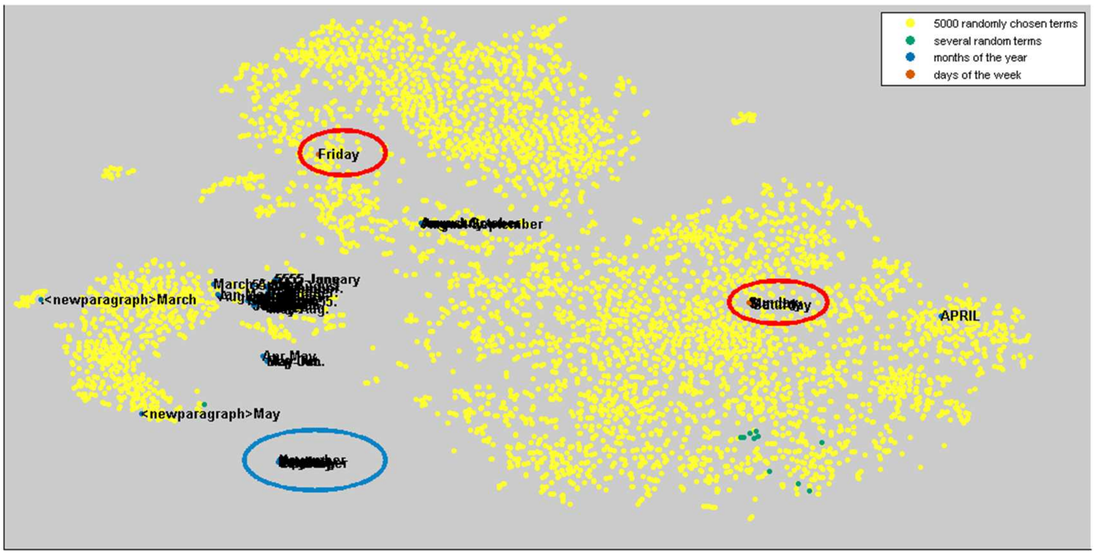


Figure 7: Isomap on the dataset with window size 1

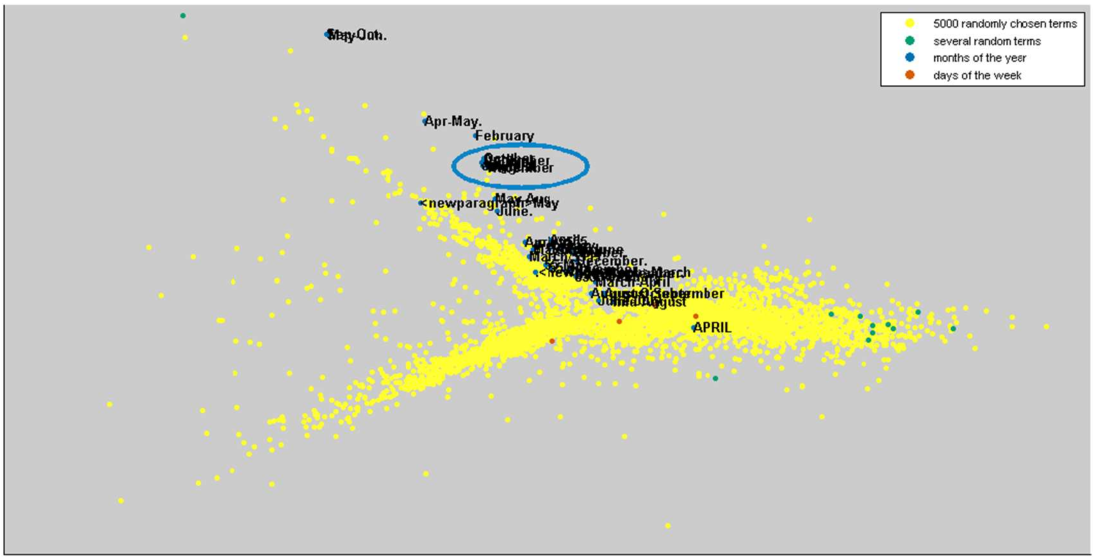
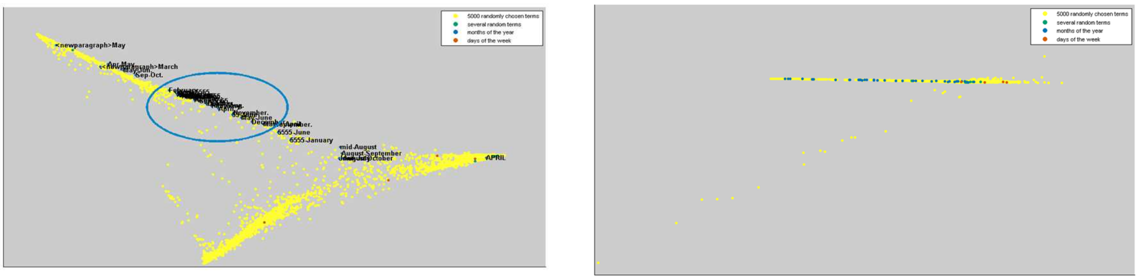
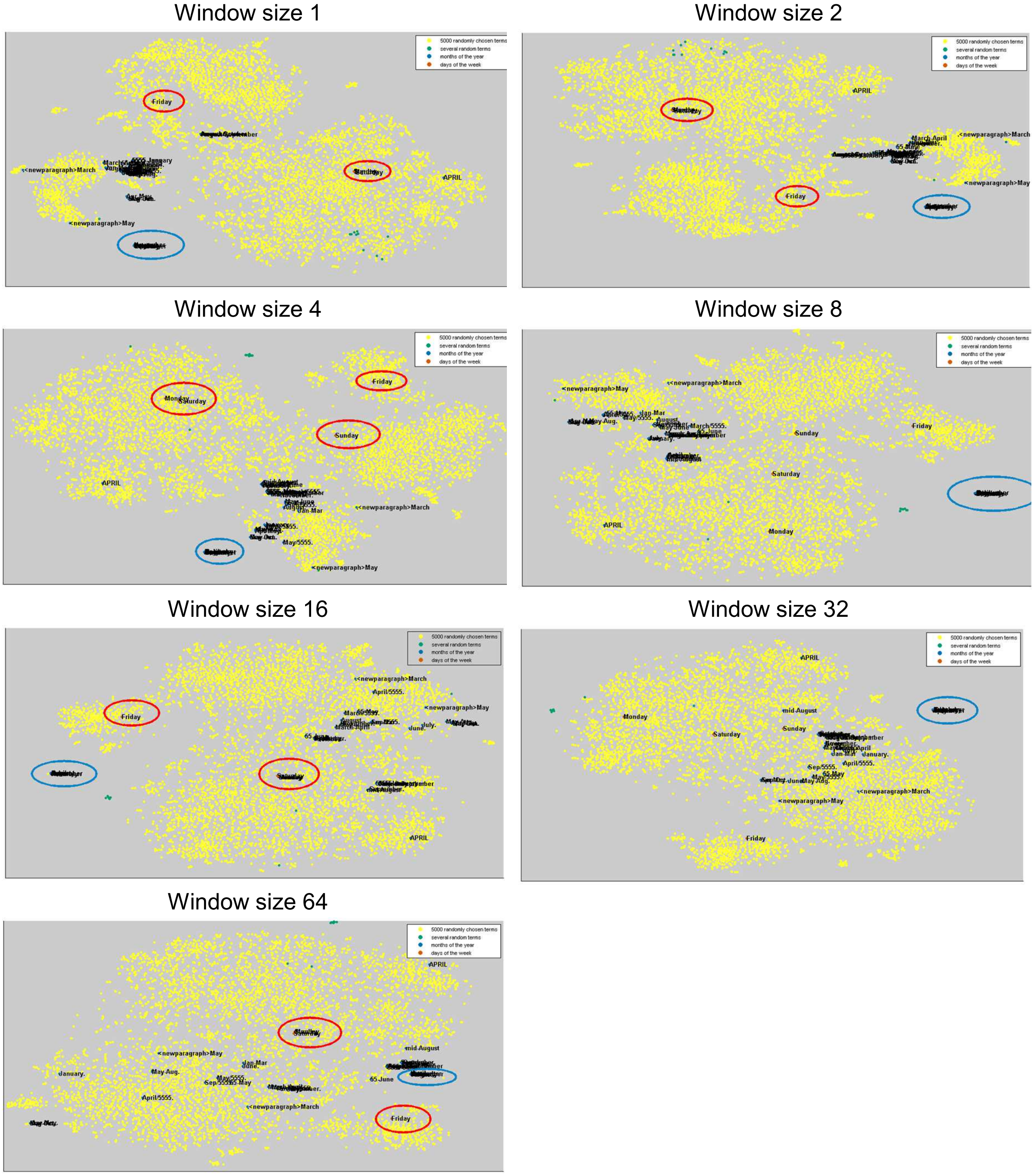


Figure 8: LE and LLE on the dataset with window size 1



Note: LE and LLE plot are on the left and right side, respectively.

Figure 9: T-SNE on seven different window sized datasets



Window size 1

Window size 2

Window size 4

Window size 8

Window size 16

Window size 32

Window size 64

Although t-SNE for the 5000 randomly-chosen-words dataset generates a number of clear clusters in the projection, it does not offer clustering information according to the positions of the data points in the map. Therefore, we apply *density-based spatial clustering of applications with noise* (DBSCAN) method to the two-dimensional data produced by the t-SNE. This provides a way of evaluating the quality of the clusters generated by t-SNE. We set the value of *MinPts* to 10 and found the optimal value of *eps* which is 4.5 in the *k*-dist graph. The result of DBSCAN with *MinPts* 10 and *eps* 4.5 is shown in figure 10. DBSCAN discovers seven clusters on the t-SNE two-dimensional map. There are two big clusters, cluster 1 and 2, which together include almost 96% of the total terms. The two big clusters are surrounded by the remaining five clusters. The details of clusters are listed in table 3. From table 3, we observe that every cluster includes terms with the similar structure or meaning with the notable exception of cluster 1 which contains the most terms. With regard to the position in figure 10, the DBSCN t-SNE map, cluster 1 and cluster 5 are closer to each other (both clusters contain the similar structure of terms, that is <newparagraph> + term, but in cluster 5 there is an extra period in front of the whole term.

Figure 10: DBSCAN on t-SNE 2-dimensional map

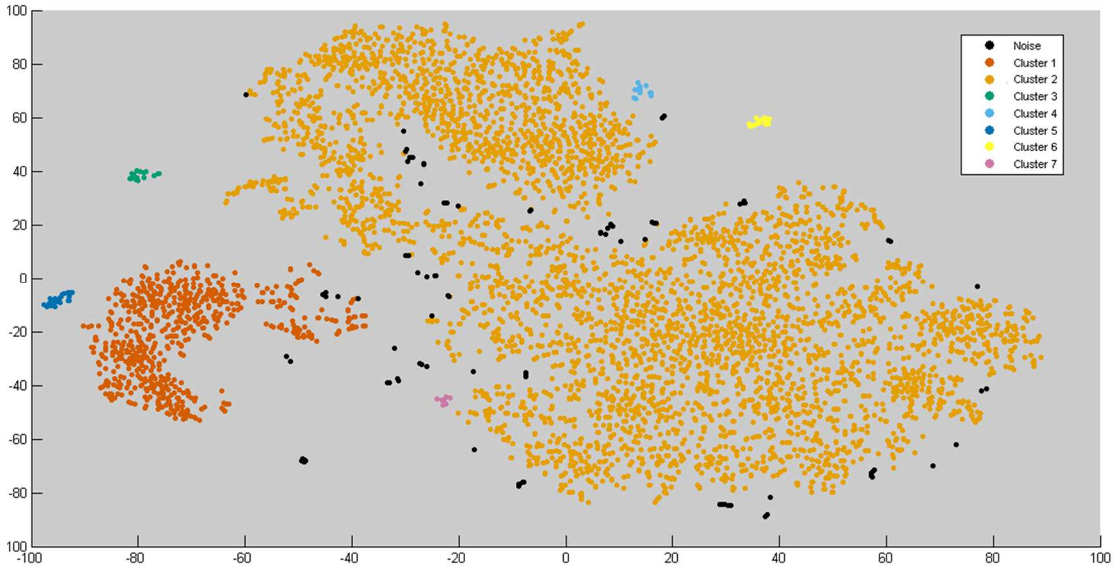


Table 3: The details of the DBSCAN clusters

Cluster	Number of terms	Type or structure of terms	Examples
Cluster 1	630	1. <newparagraph> + term 2. Terms that represent months of the year with some additions (prefix and suffix)	<newparagraph>Witt 5555-June May/5555
Cluster 2	4226	Terms from different fields	biomechanics Friday
Cluster 3	20	Combinations of two capital letters	MO WZ
Cluster 4	15	Names of people	Abbeele Erden
Cluster 5	24	Period + <newparagraph> + term	.<newparagraph>Sandler .<newparagraph>Gordon
Cluster 6	18	Combinations of two capital letters + period	AF. CB.
Cluster 7	9	Names of syndrome	Kallmann Aicardi

4.1.3 A six ‘word lists’ filtered dataset

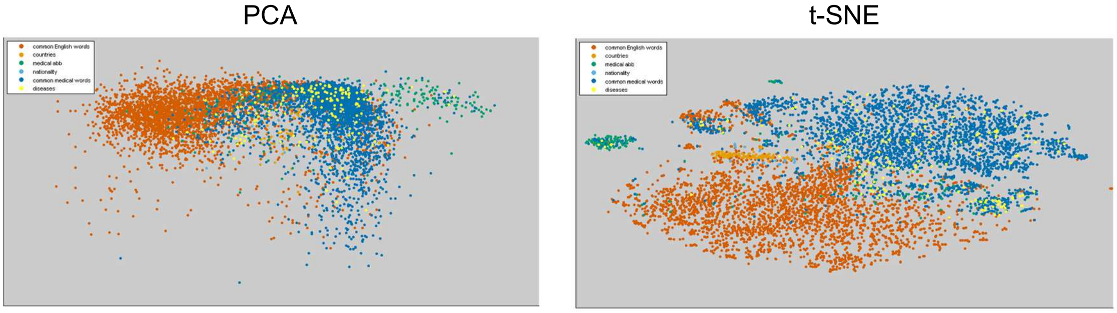
To check whether dimensionality reduction methods also work well on specific data, we apply the methods on the datasets filtered by six ‘word lists’:

(1) 6035 of the most common English words derived online (http://www.insightin.com/esl/);
(2) 201 country and continent names (https://www.countries-ofthe-world.com/all-countries.html);
(3) 490 medical abbreviations (basic medical terminology from U.S. Army medical department centre and school);
(4) 186 nationalities (https://en.wikipedia.org/wiki/Lists_of_people_by_nationality);
(5) 16456 the most common medical terms (excl. disease names) (http://www.medicinenet.com/list_of_common_medical_abbreviations_and_terminology/views.htm);
(6) 4638 of the most common disease names (http://www.medicinenet.com/diseases_and_conditions/alpha_a.htm).

The filtered dataset includes 7044 terms with vectors in which word lists 1-6 account for 48.2%, 1.7%, 4.0%, 0.4%, 43.4%, and 2.3% of 7044 terms, respectively.

Figure 11 shows the results of the DR methods in the six ‘word lists’ filtered dataset with window size 1. The results of five DR methods in other window-size datasets is similar to that in window-size 1 dataset, which is presented in figure 33 in the appendix A. In PCA plot of figure 11, the first principal component separates the groups that represent most common English words, the most common medical terms and medical abbreviations. However, if we colour all data points to black, it is hard to obtain clear group information. Compared to PCA, t-SNE plot shows some cluster patterns, which we also find in the 5000 random word experiment. In t-SNE plot, most country terms (orange dots) tend to stay close together and nationality terms (blue dots) are around the country terms. Some medical abbreviations (green dots) are clustered together. The rest of medical abbreviations and most disease name terms (yellow dots) are dispersed in the medical terms group (dark blue dots) and common English words group (red dots). We zoomed in the area where the most common English terms have an overlap with the medical terms. It is not surprising to find that the terms from the ‘most common English’ group are words that have been used a lot in medical field, like cells, transport, and metabolism. T-SNE also works well in other window-size datasets, which is shown in figure 12. Conversely, LLE produces a projection in which most data points are overlapping which renders them not useful. Almost all data points in LE are in a single line segment.

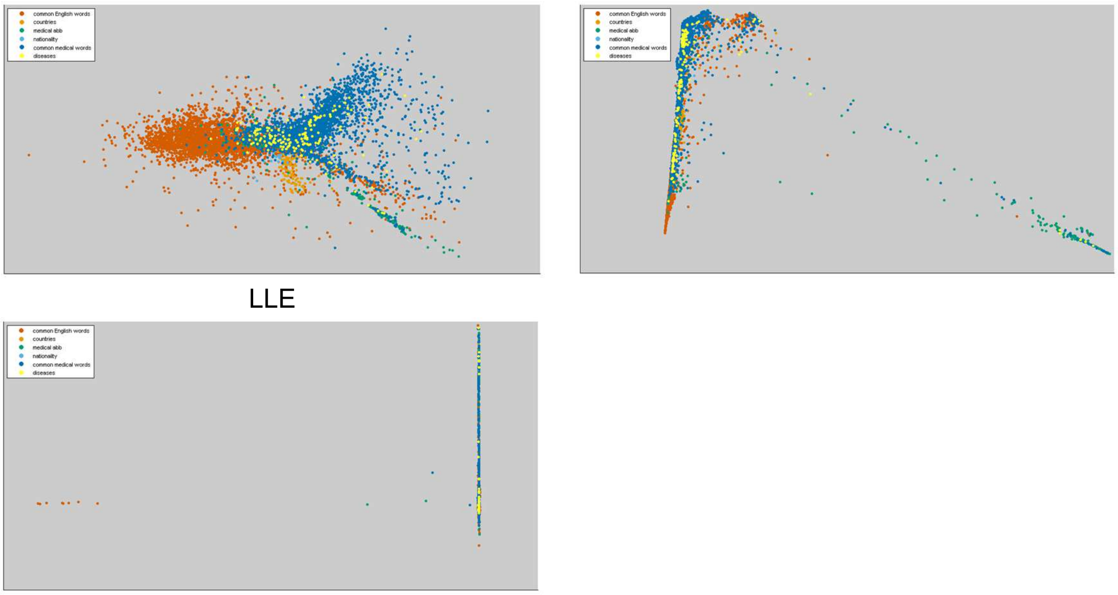
Figure 11: DR methods on six ‘word lists’ filtered dataset with window size 1


PCA

t-SNE

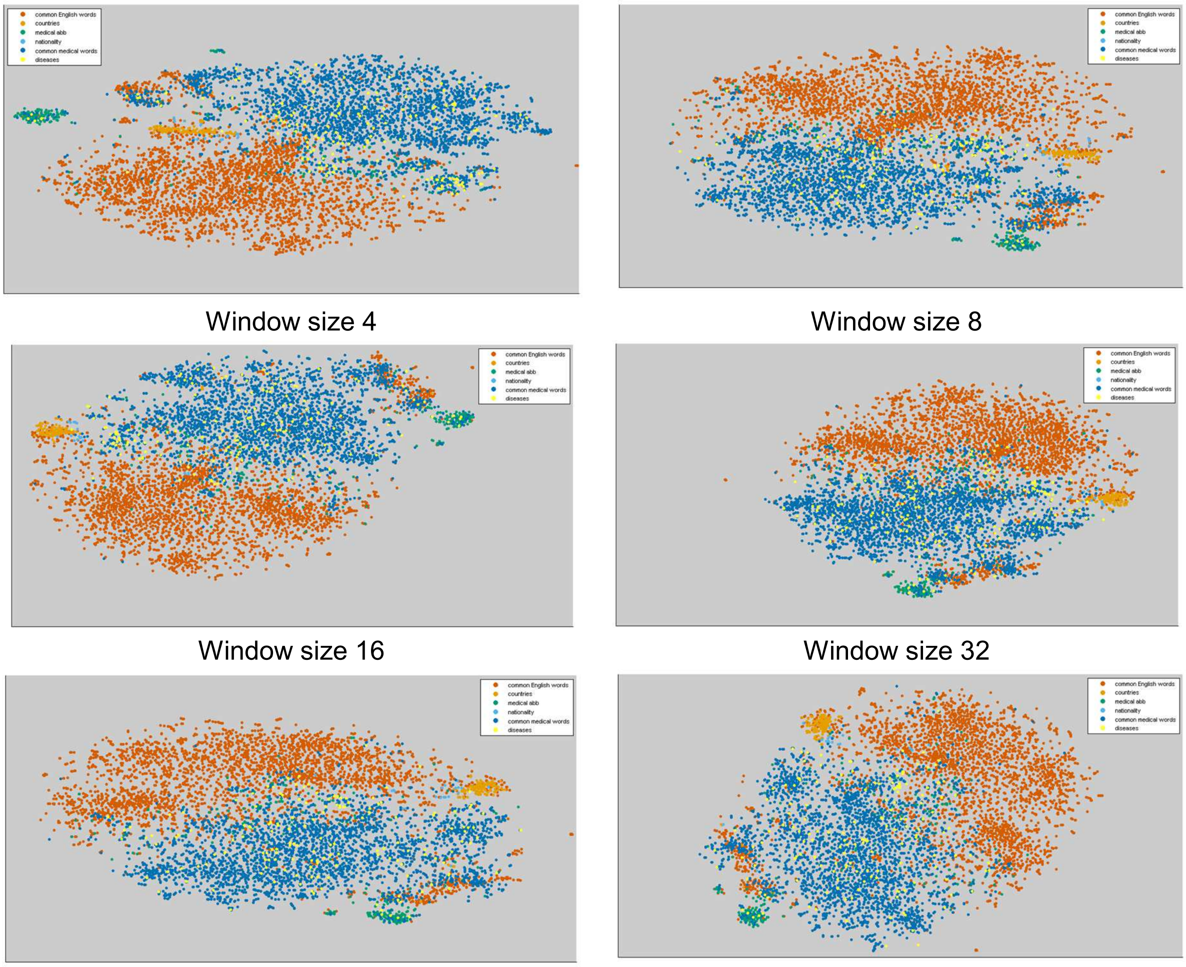
Isomap

LE


LLE

Note: we labelled data points based on the class information by colouring points that represent ‘common English words’, countries, medical abbreviations, nationalities, ‘common medical words’ and the ‘names of disease’ into red, orange, green, blue, dark blue and yellow, respectively.

Figure 12: T-SNE on the six ‘word lists’ filtered datasets with a different window size


Window size 1

Window size 2

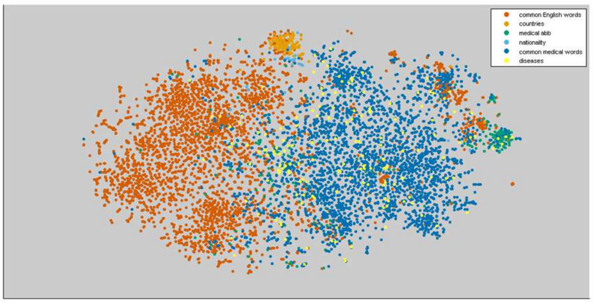
Window size 4

Window size 8

Window size 16

Window size 32

Window size 64



4.2 Stability analysis of dimensionality reduction methods

In this analysis, we evaluate the stability of the DR techniques with respect to the variations in the cost function parameters and data. Although the performance LE and LLE is unsatisfactory compared to other DR techniques, we want to check whether the variations of the cost function parameters and data will have an effect on the performance of LE and LLE. The analysis is carried out on the six-category filtered dataset. The settings of the techniques are listed in table 4.

Table 4: The settings of five dimensionality reduction methods

Technique	Parameters	settings
PCA	None	None
t-SNE	Perplexity (P): the effective number of neighbours	5<P<50
Isomap LE LLE	*k*: number of nearest neighbours	5<P<50

Source: the table is created based on the table in Carcía-Fernández (2013), p97.

4.2.1 Experimental setup

We are conducting two experiments in the study. The details of the experiments are shown below:

Experiment 1. This experiment is to study the changes of plots under the variation in the parameter, *perplexity P* or the number of nearest neighbours *k,* using the identical dataset (*N*=1000 terms). The values of *P* and *k* are 5, 10, 15, 20, 25, 35, and 45.

Experiment 2. For this experiment, we will study the performance of the DR methods on the incrementally changing size of the datasets, starting with a dataset of 1000 terms to 1500, 2000, 2500 terms.

In order to improve the stability of the DR algorithms, we apply pre- and post-processing in convex and nonconvex techniques (Carcía-Fernández et al, 2013):

1. For PCA, Isomap, LLE and LE, we use the post-processing method, *Procrustes Analysis*. Carcía-Fernández et al. (2013) propose to use Procrustes Analysis to align the shapes of projections to address this problem. Procrustes Analysis is a point-by-point shape alignment which aligns shapes to the baseline shape by re-scaling each shape to a uniform size,

translating each shape to its centroid and rotating each shape around the baseline shape until the sum of the squared distances between the corresponding points is minimised (Stegmann and Gomez, 2002; Dryden and Mardia,1998; Kendall, 1989). The projections after Procrustes Analysis help us evaluate the stability of the DR methods under the parameter and data variations efficiently and fast. In our study, we set the maps of the convex techniques with *k* equal to 5 for experiment 1 and data size equal to 2500 for experiment

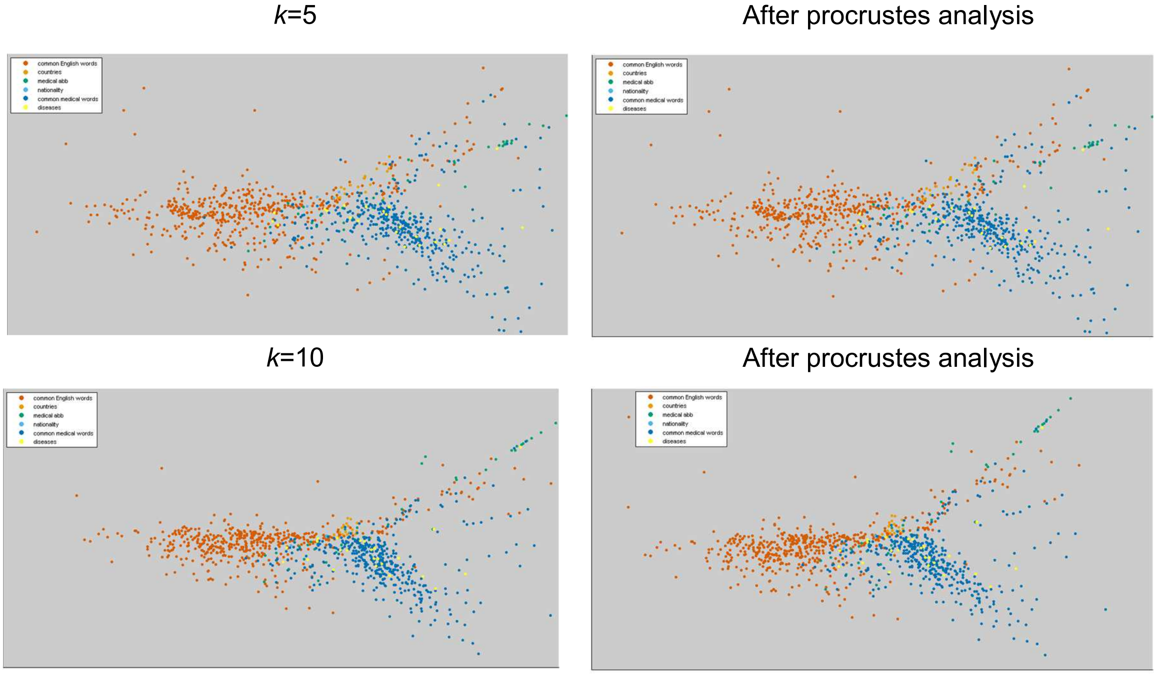
2. The nonconvex technique t-SNE initializes the algorithm randomly and data points presented in each iteration is also random. This property may impact in the stability of t-SNE, which biases the study of the stability of t-SNE under the variations in perplexity and data. To address this issue, we apply a simple pre-processing method: we fix the random seed that produces initial points to *N(0,* 10^{-4}*I)* thus controlling the randomness of the initialization of the algorithm.

4.2.2 Results of the experiment 1 and 2

Experiment 1

The results are shown in figure 13-16**.** Figure13 shows that the performance of Isomap under the variation in *k* is stable. In figure14, compared to other projections with different perplexity, data points in the projection with perplexity 5 tend to stay closer, thus generating small clusters. Although the shapes of the t-SNE projections change under the variation in perplexity, data points that stay close in one projection still cluster together in another projection. Over the perplexity's value of 5, the variation of perplexity does not bring substantial change on the quality of the embedding. It can be seen in figure 15 and 16 that the performance of LE and LLE is not improved considerably along with the change of *k* although data points in LLE are more distributed instead of in a single segment. In general, the performance of Isomap and t-SNE is relatively stable in terms of the influence of cost function parameters.

Figure 13: The performance of Isomap with *k* 5, 10, 15 and 25

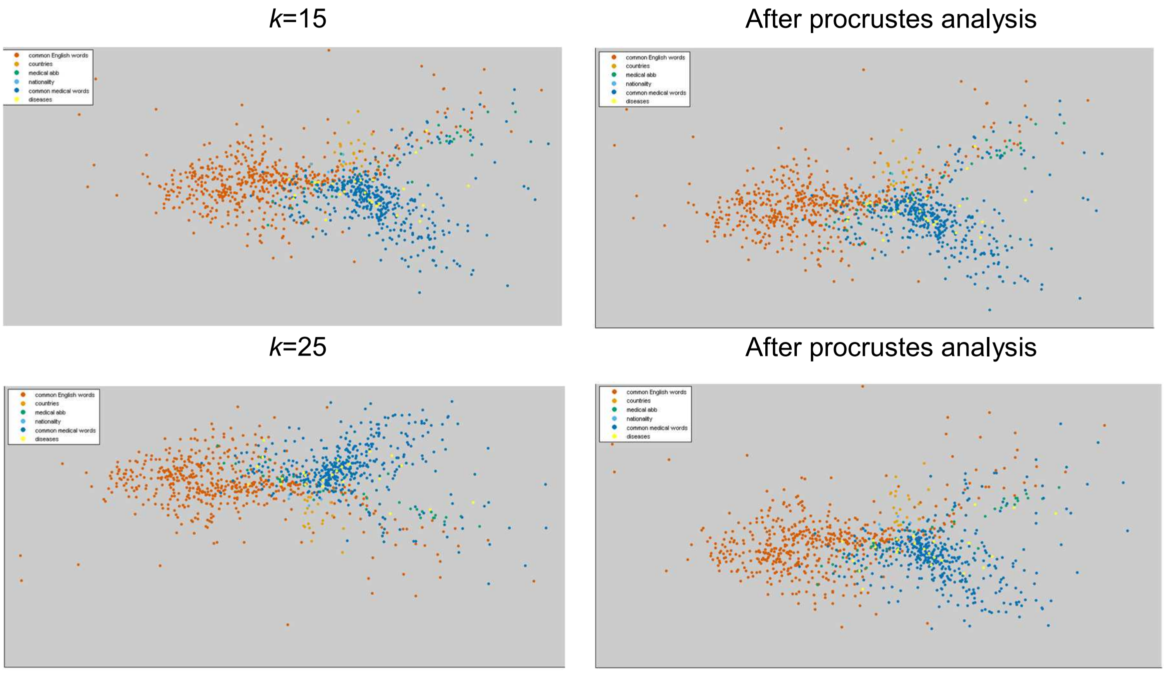


k=5

After procrustes analysis

k=10

After procrustes analysis



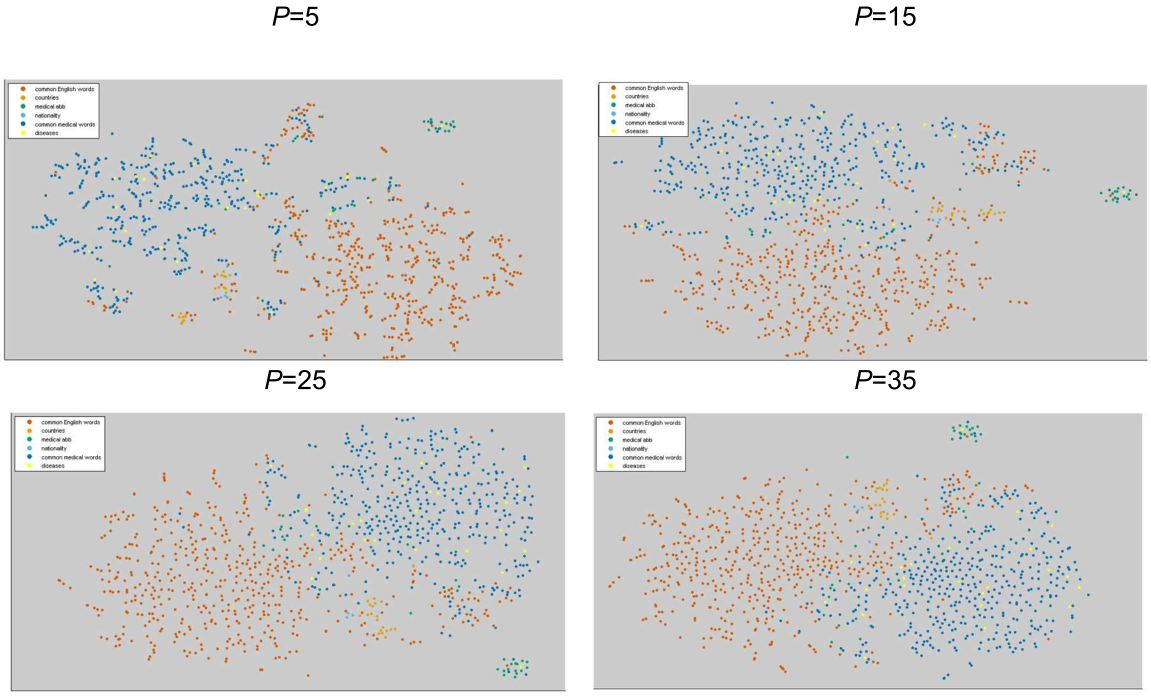
k=15

After procrustes analysis

k=25

After procrustes analysis

Figure 14: The performance of t-SNE with perplexity 5, 15, 25 and 35



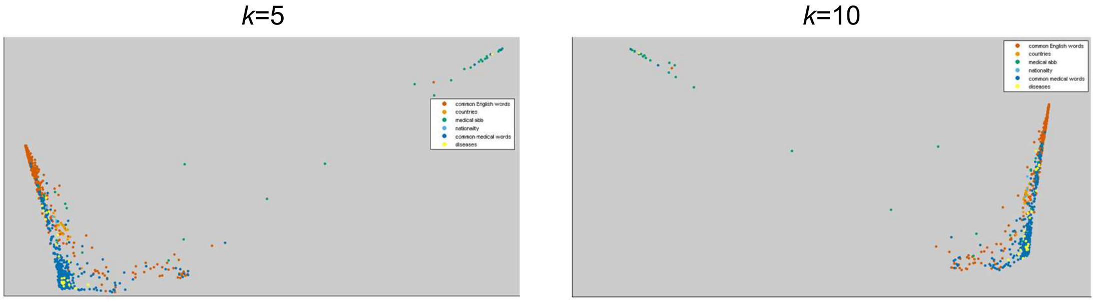
P=5

P=15

P=25

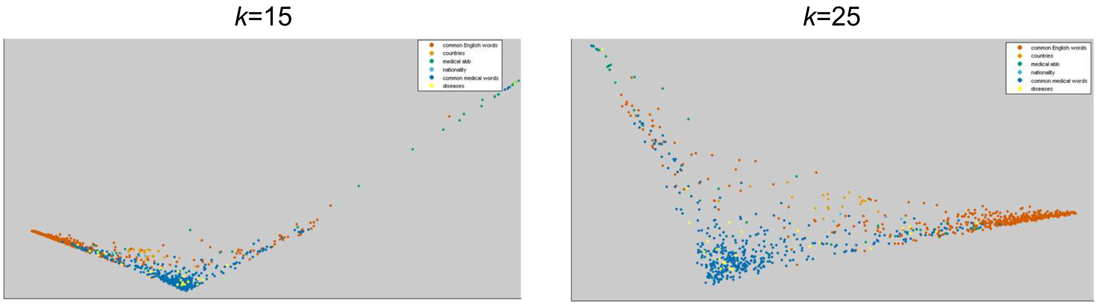
P=35

Figure 15: The performance of LE with k 5, 10, 15 and 25



k=5

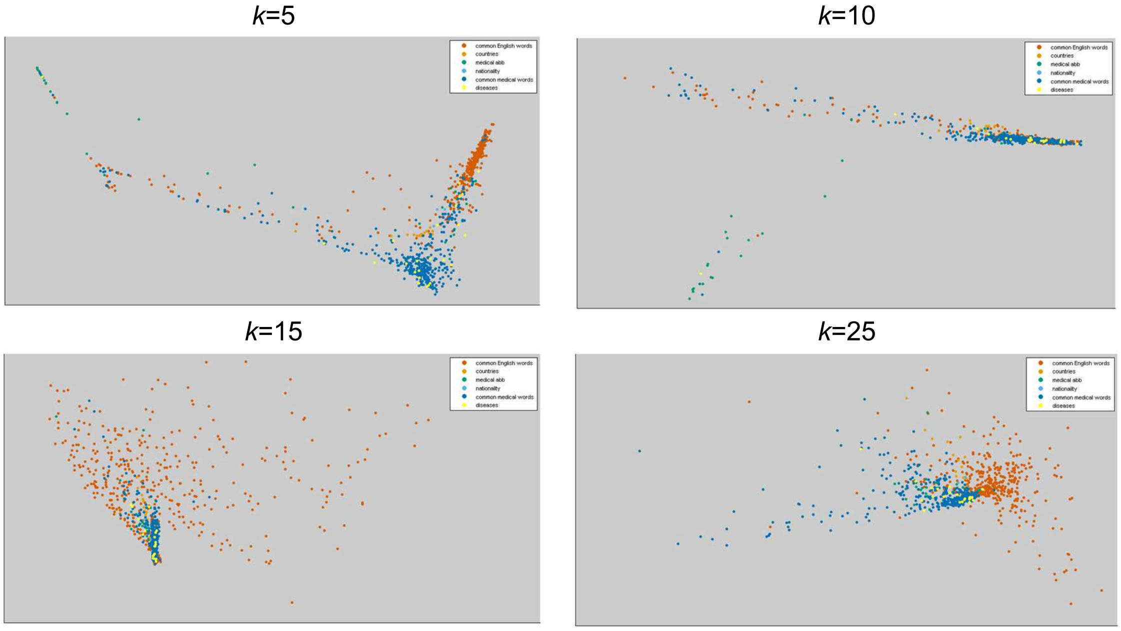
k=10



k=15

k=25

Figure 16: The performance of LLE with k 5, 10, 15 and 25



k=5

k=10

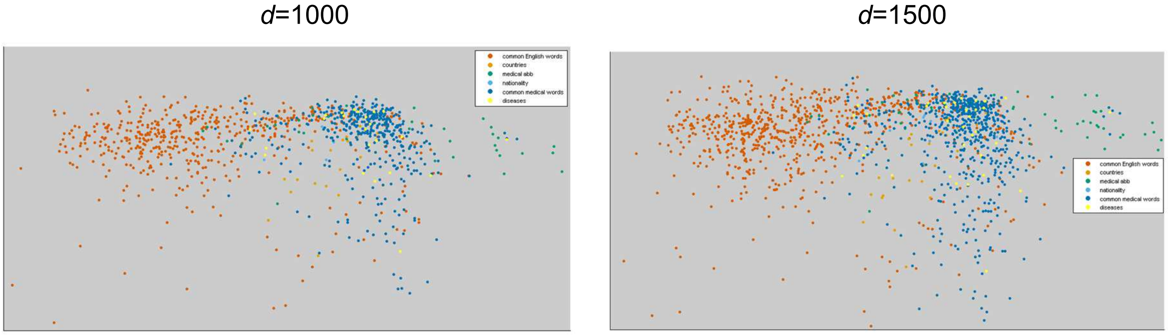
k=15

k=25

Experiment 2

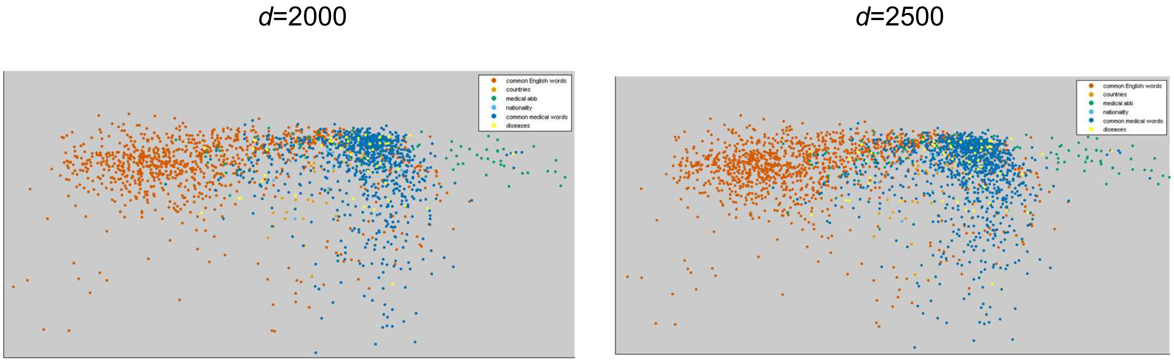
The results in figure 17-21 show that the performance of PCA and Isomap with respect to the incrementally changing size of the datasets is stable based on the similar shapes of the maps. The performance of PCA is more stable than Isomap. It confirms what we observe in experiment 1 that although the shapes of the t-SNE two-dimensional data representations vary with the change of perplexity and data size, the clustering information stays almost the same. It is important to emphasize that the quality of LE and LLE embedding is not greatly improved under the data-size variation.

Figure 17: The performance of PCA on 1000-, 1500-, 2000- and 2500-terms datasets



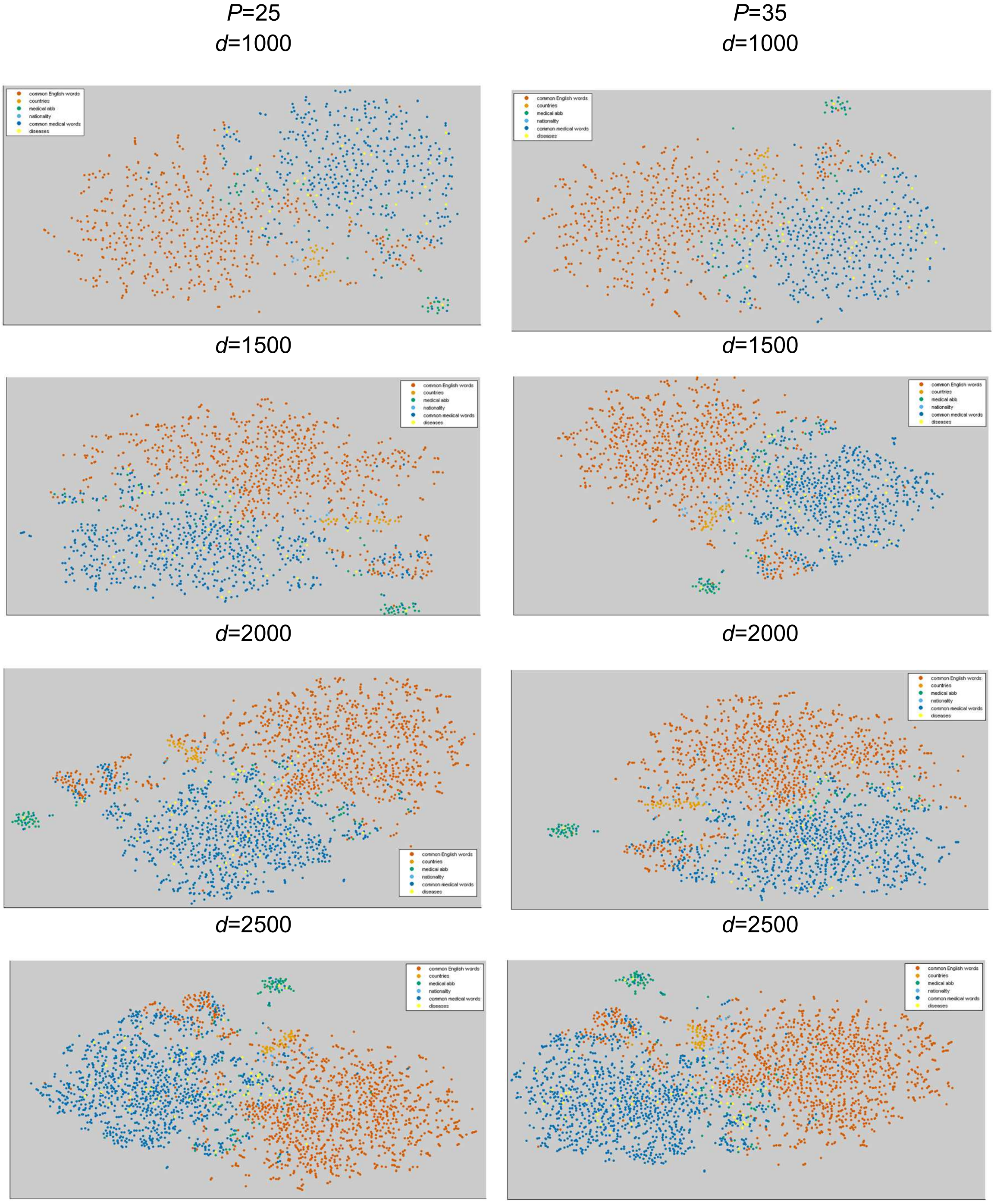
d=1000

d=1500


d=2000

d=2500

Figure 18: The performance of t-SNE on 1000-, 1500-, 2000- and 2500-terms datasets


P=25

P=35

d=1000

d=1000

d=1500

d=1500

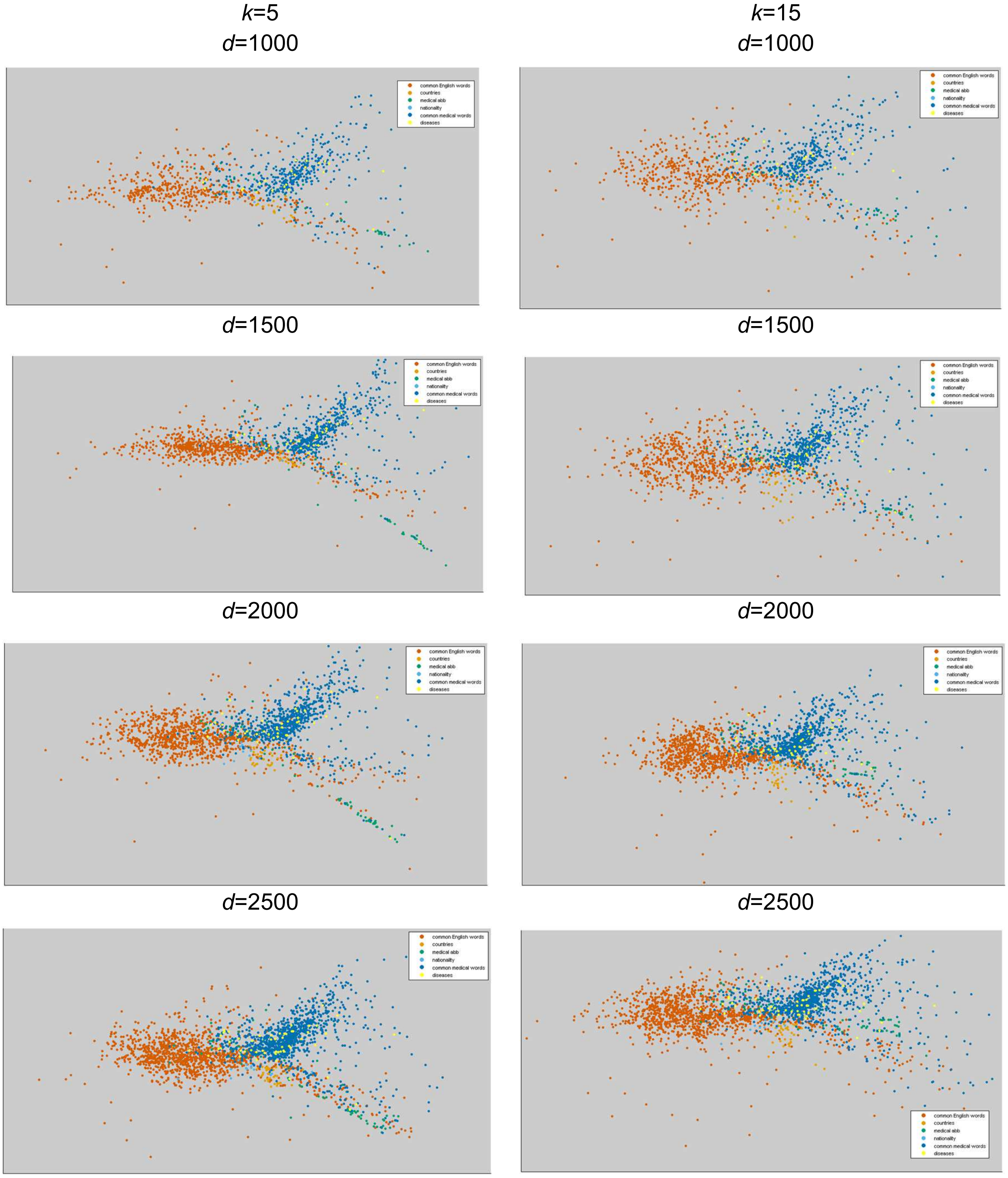
d=2000

d=2000

d=2500

d=2500

Figure 19: The performance of Isomap on 1000-, 1500-, 2000- and 2500-terms datasets



k=5

k=15

d=1000

d=1000

d=1500

d=1500

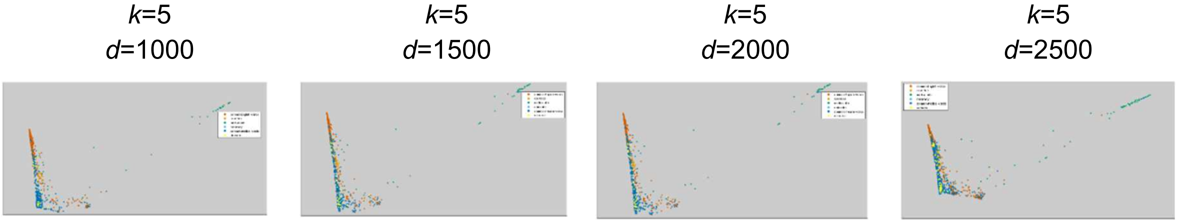
d=2000

d=2000

d=2500

d=2500

Figure 20: The performance of LE on 1000-, 1500-, 2000- and 2500-terms datasets



k=5

d=1000

k=5

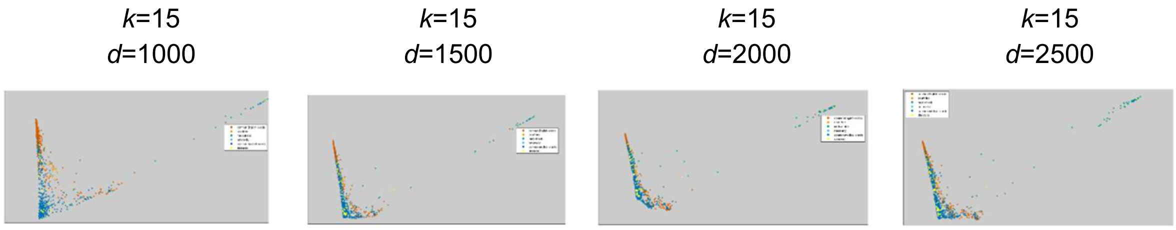
d=1500

k=5

d=2000

k=5

d=2500



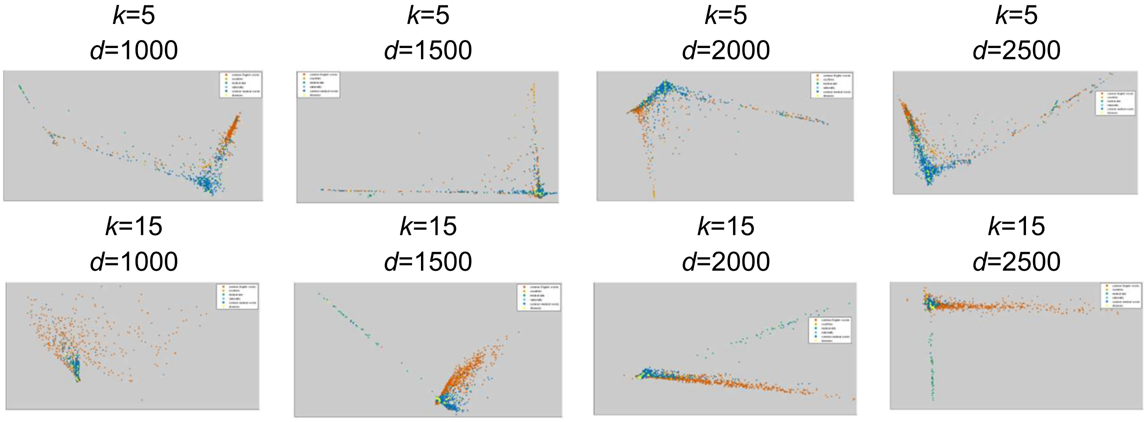
k=15 *d*=1000

k=15 *d*=1500

k=15 *d*=2000

k=15 *d*=2500

Figure 21: The performance of LLE on 1000-, 1500-, 2000- and 2500-terms datasets



k=5 *d*=1000

k=5 *d*=1500

k=5 *d*=2000

k=5 *d*=2500

k=15 *d*=1000

k=15 *d*=1500

k=15 *d*=2000

k=15 *d*=2500

4.2.3 The t-SNE clustering information analysis

In the study of the stability of the DR techniques under the variations in the cost function parameters and data, we found that, although the shapes of t-SNE projections change as perplexity and data size change, the data points in the resulting embedding which are close still stay in the map with different perplexity or data size. In other words, the clustering information is stable. To explore the details of clustering information, we carry out the following experiment: we compute the distance of each element in the dataset travelled from one perplexity plot to another successive perplexity plot. For example, assuming that there are 1000 terms, the distances of the 1000 terms are computed by calculating the Euclidean distance between the same terms travelled in perplexity5perplexity10 (P5P10), P10P15, P15P20, P20P25, P25P35, P35P45. We combine these distances together to form a 1000-term distance dataset. These are then separated into the six groups and we obtained six distance datasets which only include the distance information of elements from the same group. The same measurement is applied for other different-sized datasets. Afterwards, we apply the summary statistics, violin plots and parallel coordinates to these datasets.

Summary and descriptive statistics

Firstly, we use the summary statistics to gain a basic understanding of the distance datasets. We show the descriptive statistics of 2500-term dataset in table 5. Table 5 shows that there is some uncertainty on the movements of data points when perplexity is too small since the standard deviation of P5P10 is always the largest. As perplexity increases, the standard deviations of the distances show a downtrend, which indicates that the movements of elements belonging to the same group become stable.

Table 5: Summary statistics of 2500-term distance dataset

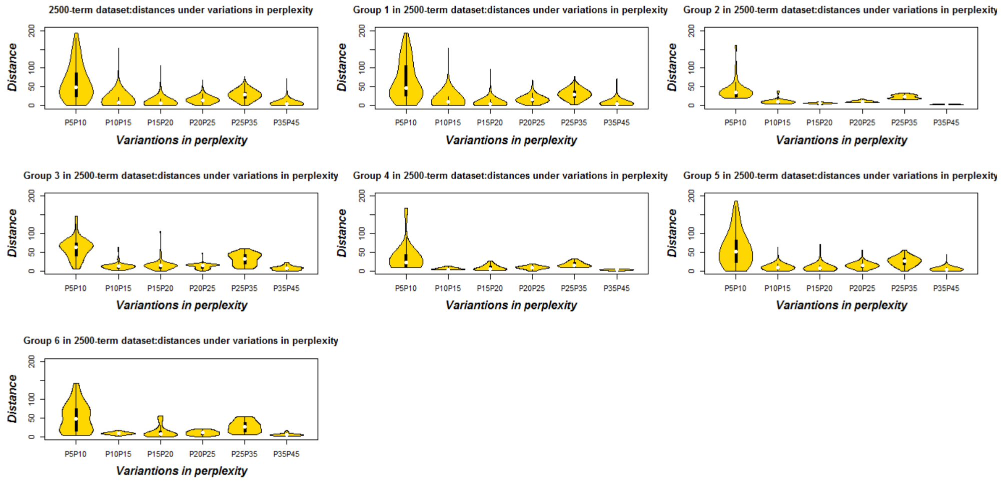
	2500 terms		Group 1		Group 2		Group 3		Group 4		Group 5		Group 6	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
P5P10	61.4	45.7	65.5	49.3	39.6	25.1	60.0	27.5	36.4	40.3	58.6	42.9	50.9	38.0
P10P15	10.7	10.3	11.6	13.4	10.7	7.0	13.3	9.1	6.3	3.2	9.6	5.1	8.9	3.3
P15P20	8.1	8.5	7.0	7.8	6.7	1.5	15.3	13.2	9.1	7.2	8.5	7.9	13.1	15.6
P20P25	15.4	7.8	16.6	9.2	10.8	2.7	13.2	6.6	9.5	4.7	14.6	5.9	10.0	5.8
P25P35	28.6	12.2	30.6	12.3	23.5	4.9	32.1	15.3	17.4	7.0	26.5	11.4	26.6	14.9
P35P45	5.9	5.3	6.5	6.3	3.6	0.5	8.1	5.6	3.3	1.5	5.1	3.7	5.1	3.3

Violin plots

Based on the previous experiment's finding that clustering patterns are stable, we expect that the distances of the elements belonging to the same class travelled between two projections with different perplexity are similar, which indicates that the distances are distributed around a certain value. Violin plots, a combination of a boxplot and a kernel density plot, are one of the most popular tools in exploring underlying distribution of data. We selected violin plots to evaluate the distribution of the distances in the dataset. Because of the limited space, we only presented the violin plot of 2500-term dataset in figure 22 and the rest violin plots from other datasets are shown in the appendix A (figure 39).

The centre white dot of the plot represents the median. The top and the bottom of the plot show the maximum value and the minimum value, respectively. The black vertical lines are the whiskers which represent the first and third quartile. The width of the plot is proportional to the estimated density. It confirms what we observed in the summary statistics part, i.e. that the standard deviation of P5P10 is always the largest. In addition, we also find that the distributions of the distances in group 1 and group 5 plots (the number of elements in group 1 and group 5 accounts for 49.2% and 42.8% of the total number, 2500 terms, respectively) are similar to the distributions in the 2500-term plot. Compared to the 2500-term plot, the travelled distances in group 2 or group 4 (except P5P10) are similar since the distributions are flat. Although the distributions in group 5 and 6 are not flat, there exists density difference in the distributions. For example, some distributions become wider at the middle. This means that most distances exhibit a similar trend. In general, distances in small groups (group 2, 3, 4 and 6) tend to be similar.

Figure 22: Violin plots of the 2500-term dataset

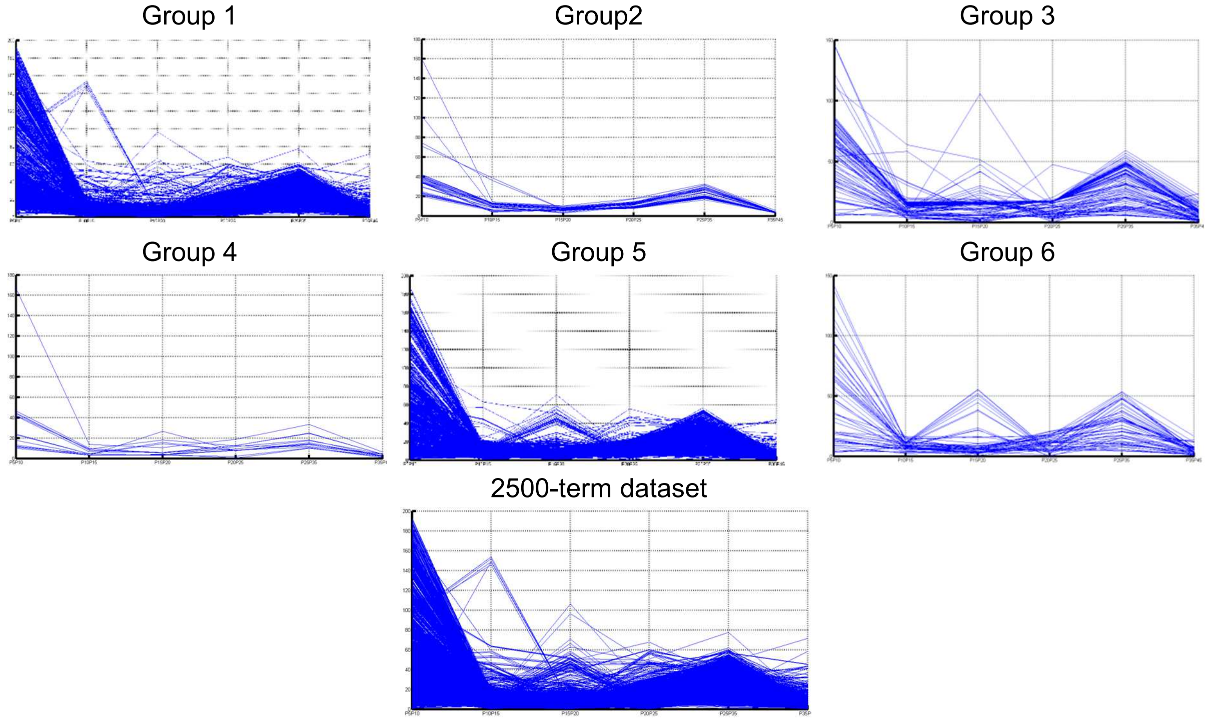


Parallel Coordinates

Parallel coordinates are an interactive visualization technique that reveals meaningful patterns of multivariate datasets. We expect that the travelled distances of elements from the same group follow the same trend in a parallel coordinates graph. Although it is difficult to dig into the details of each word in a parallel coordinates graph, it offers us a way to evaluate the trend of the distances. Because of space limitations, we only represented parallel coordinates of the 2500-term dataset in figure 23.

Figure lines are used to encode the changes of term distances along the variations in perplexity. Six vertical axes are laid out in parallel from left to right along the *x*-axis and each vertical axis represents a perplexity comparison such as P5P10. The variable names appear on the bottom of x-axis. Within a single line in a parallel coordinates display, a series of distance values belonging to a same term are connected together, in which each value is associated with a different perplexity comparison. The slopes' up and down movement of the lines display a distance trend along perplexity from one value to another. In the graphs of group 2 and 4, looking at the axes from the left to the right one can see that the movements of terms under the variations in perplexity have a similar trend since the lines are overlapping. The graphs of group 3 and 6 also reveal that most of the lines starting from P10P15 to P35P45 are overlapping at the beginning but separate at P25P35 and get closer again at P35P45. Parallel coordinates also confirm that distances in small groups (group 2, 3, 4 and 6) tend to be similar.

Figure 23: Parallel coordinates of the 2500-term dataset



Group 1

Group2

Group 3

Group 4

Group 5

Group 6

2500-term dataset

4.3 Results of intrinsic word vector evaluation

To measure the quality of the word vectors, we choose two syntactic tests and one semantic test. The details of the tests are shown in table 6. Before conducting these tests on the word vectors of medical terms, we projected the 50-dimensional skip-gram processed vectors of seven different window-size datasets each to a two-dimensional map by using t-SNE with perplexity 30.

Table 6: Examples of the syntactic and semantic tests

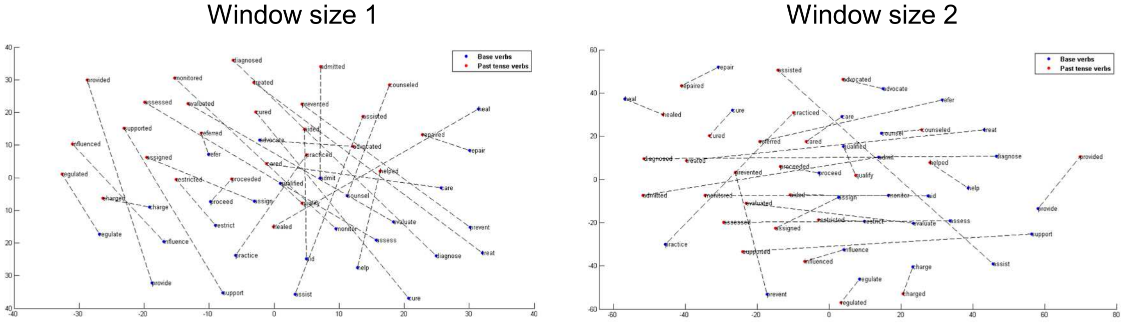
Type of relationship	Word Pair 1		Word Pair 2	
Past tense	admit	admitted	provide	provided
Plural nouns	disease	diseases	doctor	doctors
Organ and related diseases	skin	acne	heart	stroke

4.3.1 Syntactic test

Past tense

Firstly, we evaluate whether the model captures the tense pattern. We pick 27 pairs of the verbs with present and past tense from the skip-gram processed medical dataset. The results of past tense test are shown in figure 24. Blue and red dots represent present and past tense verbs, respectively. In window size 1 plot of figure 24, the base verbs are located at the relatively low part of the plot. The line segments from base verbs to past tense verbs are pointing upwards. In window size 2 of figure24, the base verbs are located at the right part of the plot. The line segments from base verbs to past tense verbs are pointing left. Compared to window size 1, the corresponding present and past tense verbs (for instance, *provide and provided*) in other window-size plots become closer. To confirm the result in the figure 24, we use "the relative distance" which is computed by dividing Euclidean distance between the selected verb with present and past tense by the mean of Euclidean distances between the particular verb and all other verbs[10]. We find that compared to window size 1, the median of "the relative distance" decreases in figure 25, this shows that the corresponding present and past tense verbs in other window sizes get closer vis-à-vis window size 1. Figure 26 shows the details of the comparison of other window sizes and window size 1 in "the relative distance". Most points are below diagonal line which confirms that "the relative distance" in other window sizes relative to window size 1 decreases. Especially, the magnitude of the decrease of "the relative distance" of verbs in window size 32 is highest. In fact, in this case 24 out of 27 of "the relative distance" of verbs decrease- this can be mathematically seen in more detail in table 8 in the appendix A. Although the results show that the corresponding present and past base verbs stay close, it does not clearly depict the special pattern found in some studies that word vectors go in a similar direction and sometimes even the length of the vectors are similar.

Figure 24: Past tense test on seven different window sized datasets



Window size 1

Window size 2

[10] All other verbs: contain both past and present tense

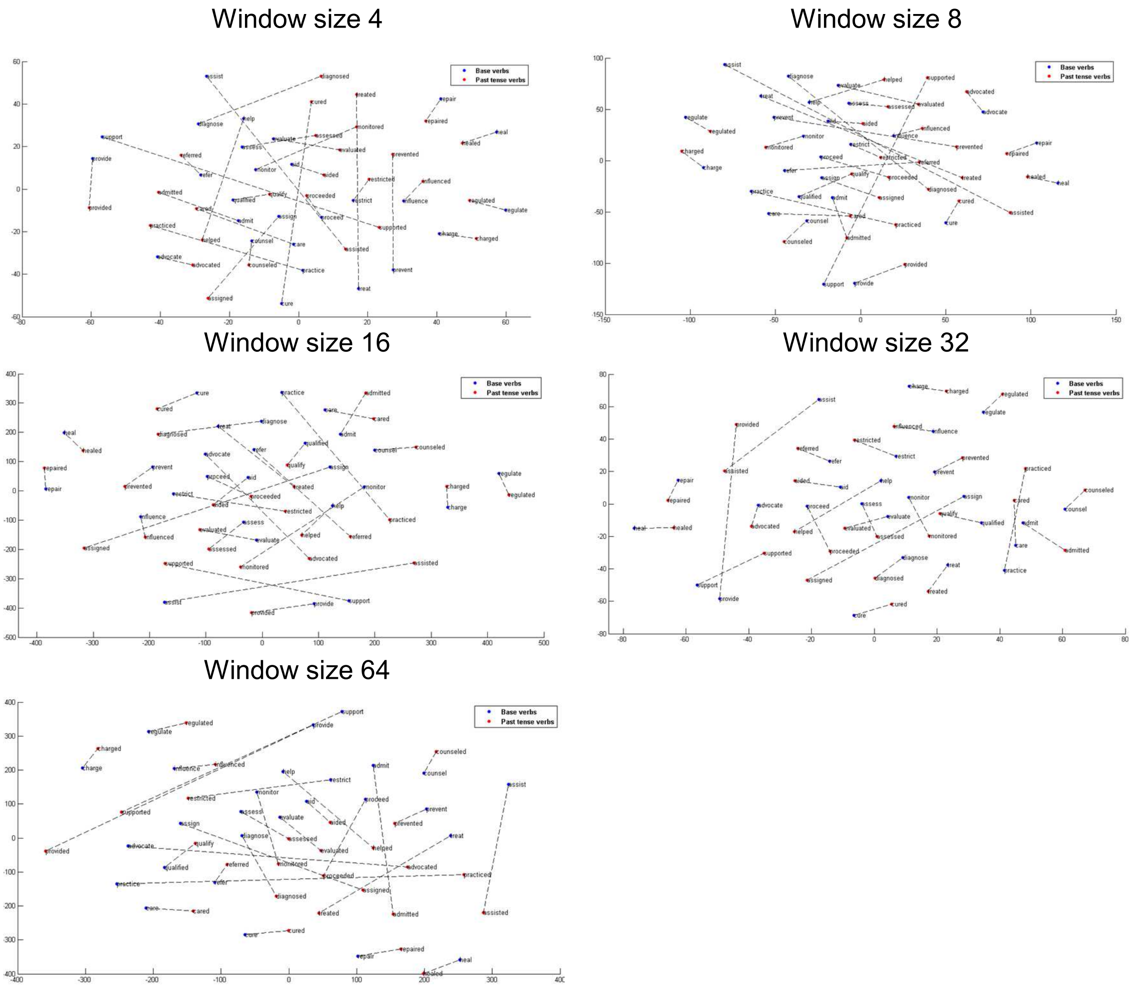


Figure 25: The median of relative distance by window size

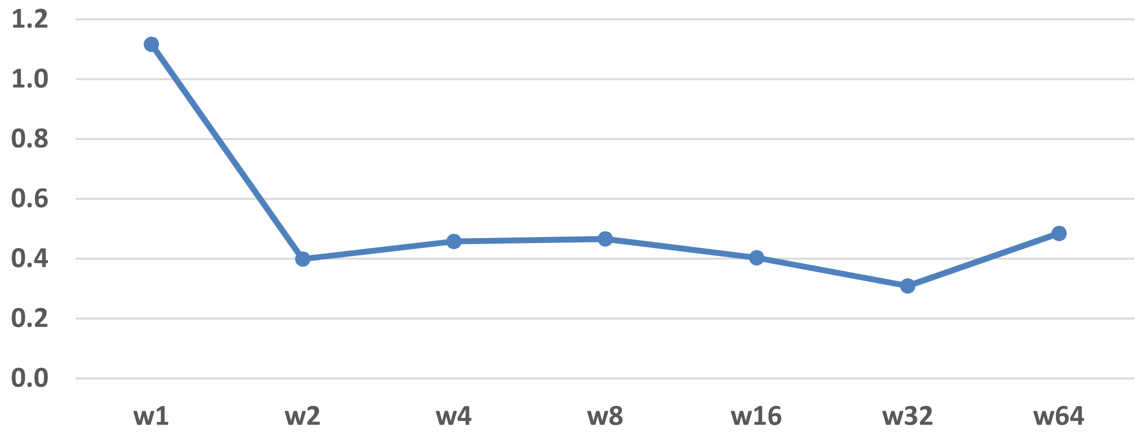
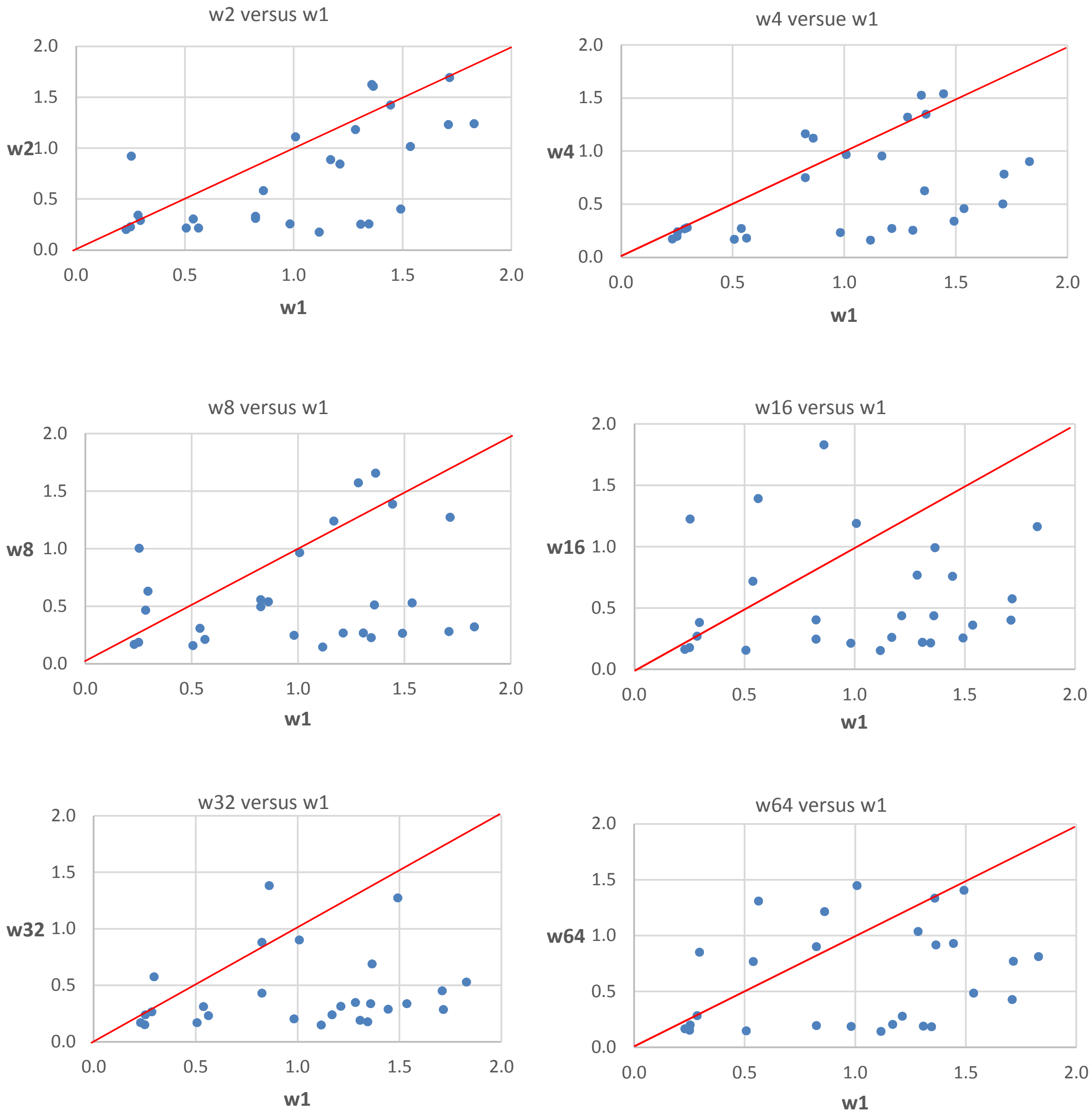


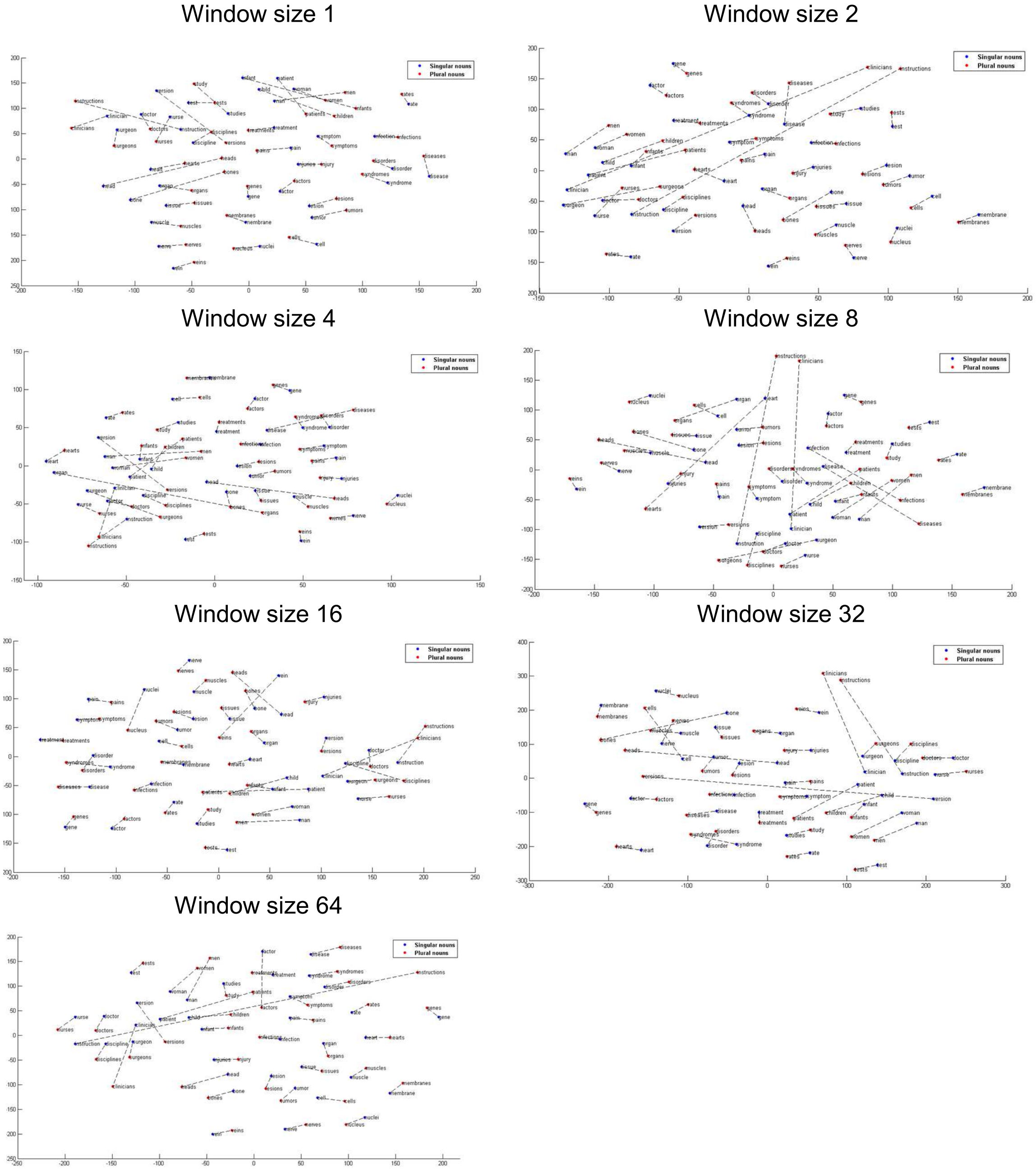
Figure 26: The scatter plots of relative distance by window size



Plural nouns

Secondly, we evaluate whether the model identifies singular and plural pattern. Figure 27 depicts the results of our analysis. Blue and red dots are singular nouns plural form of nouns, respectively. Instead of all plural nouns or singular nouns group together, singular and plural nouns with the same meaning are situated close to each other. After applying the same operation of “the relative distance” in *Plural nouns* test, we find that compared to window size 1, the magnitude of the decrease of “the relative distance” of nouns in window size 16 is higher than other window sizes. Specifically, for 26 out of 38 the “the relative distance” decrease, which you can find in table 7. It means that in term of clustering structure of the corresponding nouns in singular and plural form, window size 16 performs the best. In addition, we also observe that nouns that represent a person are staying closer.

Figure 27: Plural nouns test on seven different window sized datasets



Window size 1

Window size 2

Window size 4

Window size 8

Window size 16

Window size 32

Window size 64

Table 7: The relative distance test for plural words

nouns	w2 vs w1	w4 vs w1	w8 vs w1	w16 vs w1	w32 vs w1	w64 vs w1
bone-bones	-0.13	-0.33	-0.12	-0.37	0.12	-0.41
cell-cells	0.01	0.01	0.06	0.05	0.63	0.03
child-children	-0.11	-0.12	-0.09	-0.04	-0.10	-0.20
clinician-clinicians	1.62	0.51	1.85	0.55	1.22	0.53
discipline-disciplines	-0.02	0.00	0.17	0.20	0.02	-0.04
disease-diseases	0.32	0.45	0.91	-0.03	-0.01	-0.05
disorder-disorders	0.02	0.02	0.09	0.04	0.03	0.03
doctor-doctors	-0.01	-0.02	-0.04	-0.02	-0.05	-0.03
factor-factors	-0.07	-0.05	-0.04	-0.07	-0.05	0.38

gene-genes	-0.06	-0.05	-0.04	-0.06	-0.05	-0.05
head-heads	-0.32	0.37	0.15	-0.07	0.41	-0.31
heart-hearts	0.04	-0.07	1.16	-0.05	-0.12	-0.07
infant-infants	-0.43	-0.41	-0.43	-0.44	-0.44	-0.43
infection-infections	0.02	0.02	0.90	0.02	0.04	0.04
injury-injuries	-0.01	-0.04	-0.02	-0.05	0.00	0.01
instruction-instructions	1.35	-0.40	1.24	-0.45	0.44	1.01
lesion-lesions	-0.02	0.01	0.00	-0.01	0.01	-0.01
man-men	-0.08	0.19	0.18	-0.06	-0.15	0.14
membrane-membranes	-0.04	-0.03	-0.02	0.05	-0.03	-0.01
muscle-muscles	0.00	0.02	0.01	0.00	0.02	-0.01
nerve-nerves	0.00	0.00	0.01	-0.01	0.19	0.00
nuclei-nucleus	0.02	-0.02	-0.01	0.31	-0.02	-0.01
nurse-nurses	-0.03	-0.18	-0.18	-0.17	-0.18	-0.19
organ-organs	0.02	0.98	0.21	-0.02	-0.04	-0.03
pain-pains	-0.07	-0.10	-0.08	-0.13	-0.07	-0.07
patient-patients	0.14	0.31	0.37	0.29	0.23	0.31
rate-rates	-0.01	0.01	0.00	0.04	0.01	0.04
study-studies	-0.33	-0.30	-0.31	-0.31	-0.33	-0.33
surgeon-surgeons	0.26	0.35	0.40	-0.03	0.00	-0.01
symptom-symptoms	0.01	-0.01	0.01	-0.05	0.02	0.00
syndrome-syndromes	0.02	-0.01	0.08	0.13	0.10	0.01
test-tests	-0.01	-0.03	-0.04	-0.04	-0.05	-0.04
tissue-tissues	0.01	0.01	0.00	0.00	-0.03	-0.01
treatment-treatments	-0.04	-0.03	-0.02	-0.07	-0.02	-0.05
tumor-tumors	-0.05	0.00	-0.03	0.00	0.02	-0.03
vein-veins	0.01	0.00	0.01	0.58	0.03	0.00
version-versions	-0.45	0.16	-0.49	-0.50	0.77	-0.18
woman-women	0.01	0.30	0.39	0.06	0.18	0.12
the number of decreased "the relative distance"	20	19	19	26	17	25

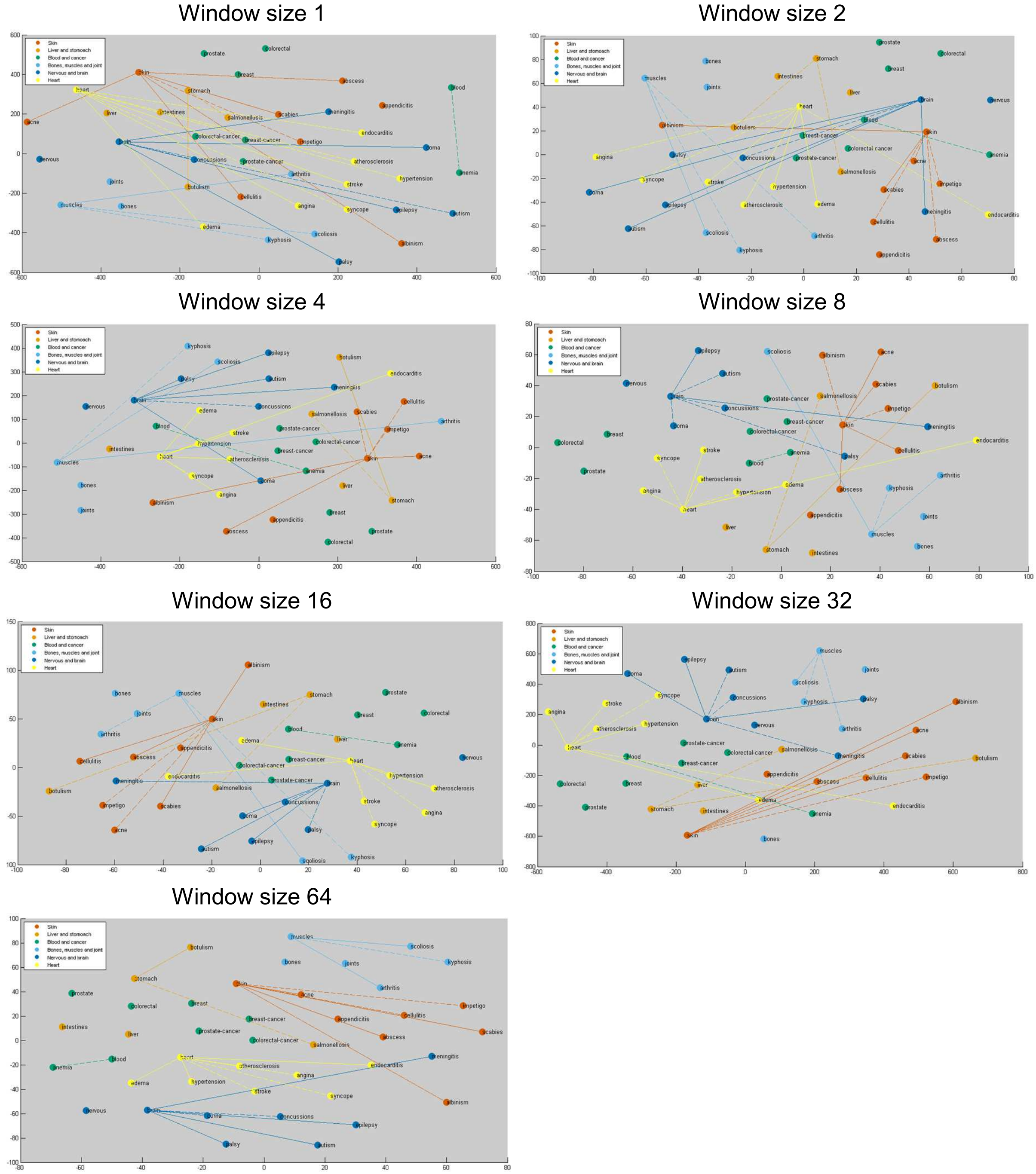
Note: w1, w2, w4, w8, w16, w32, w64 correspond to window size 1,2,4,8,16,32,64, respectively. The values are computed by subtracting “the relative distance” of window size 1 from “the relative distance” of the listed window sizes. We highlighted the negative values which is obtained when compared to window size 1, i.e. “the relative distance” of the pair of nouns dose not decrease to this particular window-size for this element.

4.3.2 Semantic test

Organ and related diseases

Thirdly, we check whether the model identifies a clear relationship between disease and the related organ. In the plots of window size 1 and 2 in figure 28, the line segments are overlapping, but you can still find that most diseases stay close to each other. In all plots, the names of cancer are close. As window size increases, diseases related to the same organ become closer but more apart from other diseases.

Figure 28: Organ and related diseases test on seven different window sized datasets



Window size 1

Window size 2

Window size 4

Window size 8

Window size 16

Window size 32

Window size 64

5 Conclusion

Nowadays, as the size and complexity of unstructured medical reports is constantly increasing, a large amount of valuable medical information is buried in these reports which may have significant applications in drug development (ACCUMULATE, 2016). In order to extract crucial medical information from various medical texts, researchers have proposed to use natural language processing. It helps to find an appropriate model that embeds terms into dense, real valued vectors that capture semantic and syntactic properties of the terms (Turian et al., 2010). However, word vectors produced by natural language techniques are always high-dimensional. Therefore, it is necessary to find an appropriate dimensionality reduction method to transform high-dimensional datasets into a useful two- or three-dimensional space thus providing a way of visualizing the datasets with the naked eye. In this study, we introduce five dimensionality reduction methods to the skip-gram processed medical dataset and compare the performance of these DR methods.

Although t-SNE generates a number of clear clusters in the projection, it does not offer clustering information according to the positions of the data points in the map. Therefore, we need to find a proper clustering technique for our study to obtain details on the clusters which enables us to estimate the quality of clusters. With the application of clustering techniques to large spatial datasets becoming increasingly common, researchers pay more attention to the clustering techniques that combine two requirements (Ester et al., 1996): 1) determination of input parameters for a specific database with less domain knowledge; 2) discovery of clusters of arbitrary shapes. Although there are a variety of clustering techniques, few techniques can combine the two requirements at the same time. For instance, partitioning algorithms like k-means have difficulty in dealing with the shapes of clusters that are non-spherical, while hierarchical algorithms do not have a clear criterion to find an appropriate termination parameter (Tan et al., 2006). DBSCAN is a clustering technique which satisfies both requirements. In consideration of arbitrarily-shaped clusters in the t-SNE projection, we decide to use DBSCAN to the t-SNE two-dimensional map to explore the details of the clusters. The results of DBSCAN show that data points in the same group have either similar structure or similar meaning, which confirms that the clusters generated by t-SNE are meaningful. One of major weaknesses of the DBSCAN is its assumption that the distribution of the points within each cluster is uniform (Trikha and Vijendra, 2013). Due to this assumption, DBSCAN might ignore some meaningful clusters in our t-SNE map. Fang et al. (2014) and Elbatta and Ashour (2013) improve the DBSCAN algorithm with respect to the uniform density assumption by introducing several values of parameter Eps, which can be used in future research.

Although Isomap, LLE and LE perform dimensionality reduction by attempting to preserve pairwise geodesic distances over a manifold while PCA tends to retain pairwise Euclidean distances, there is the common characteristic among them that the low-dimensional representations y_i are obtained by performing an eigendecomposition of a pairwise Euclidean or geodesic distance matrix (Van der Maaten et al., 2008b). Because of the algorithm, indeterminacies will be brought to representations, thus causing irrelevant geometric transformations like mirroring, rotation and translation of the projections (Carcía-Fernández et al., 2013). We find that these intrinsic geometric transformations slow down the comparison of the geometric variations of the projections. In the study, we set the shapes of the convex techniques with k equal to 5 and data size equal to 2500 as the baseline shape and considered the geometric variations of projections after Procrustes Analysis - these variations are caused by parameter and data change. In our study, we find that the projections after Procrustes

Analysis help us evaluate the stability of the DR methods under the parameter and data variations efficiently and fast.

In our study, t-SNE outperforms PCA, which can be explained by the PCA property of finding linear representations of the original data. Although t-SNE and Isomap are sharing many advantages as a nonlinear technique, our results reveal that the performance of Isomap is inferior to t-SNE. We think that it is partly due to the sensitivity of Isomap to short-circuiting and the focus of the Isomap on retaining large geodesic distance (Tenenbaum et al., 2000; Van der Maaten et al., 2008b). The second reason is confirmed by the similar projections of Isomap and PCA in the study.

In our study, LE and LLE do not yield satisfactory performance on transforming the high-dimensional data to a useful two-dimensional map. Potentially this may be due to three reasons: Firstly, based on the work of Van der Maaten et al. (2008b), we can surmise that since LE and LLE use a simple covariance constraint to avoid the trivial solution, that is all data points collapsed onto a single point. Some undesired embeddings can easily meet the constraint. For instance, most data points are embedded in a single line segment and a few points are scattered, which we find in our study. Secondly, LE and LLE need to find the smallest eigenvalues to obtain embedding coordinates Y. In practice, it is difficult to identify the smallest eigenvalues because of the extremely small values of these eigenvalues. Third, the assumption of LLE that the local structures of the manifold are linear requires the manifold to be smooth, which may be not satisfied in our dataset.

Overall, t-SNE works the best for our analysis in terms of the cluster structure and the stability performance.

In our analogy test study, the t-SNE two-dimensional projections reveal that similar words are close to each other and the clustering patterns change under the window-size variation. However, word vectors do not clearly present the syntactic and semantic relationships between two words in these three analogy tests. In this relationship, word vectors go in a similar direction and sometimes even the length of the vectors are similar. This may be due to the following reasons:

1. The size of our training data (147,764 words) is small. Word2vec requires a large size of training data to obtain meaningful word vectors. Mikolov et al. (2013a; 2013b) use a dataset with about one million words and a dataset with about 33 billion words for the single-word and the phrase analogy task, respectively. Because the medical knowledge is complex and medical text is non-well-formed, researchers also apply word2vec to a large amount of medical corpora (Miñarro-Giménez et al., 2015; Muneeb et al., 2015; Choi et al., 2016). Miñarro-Giménez et al. (2015) obtained an accuracy of 43.9% that captures the *may_treat* relationship based on the corpus with about 234 million words.
2. Our training data is unprocessed. Due to the fact that word2vec does not perform term normalisation before word embedding, the words with the same meaning but different structure (e.g. 'disease' and 'disease.') may be treated as different words, thus influencing the quality of word vectors. Miñarro-Giménez et al. (2015) confirms that the pre-processing on the original data, for instance removing all punctuation signs and transforming all words to lower-case, increases the quality of word vectors.
3. The dimensionality of word vectors (50-dimensional vectors) may not be big enough. Vector dimensionality has a positive impact in the accuracy of word vectors in analogy tests (Miñarro-

Giménez et al., 2015; Mikolov et al., 2013a). In terms of unstructured form and complexity medical knowledge, a higher word vector may work better for our medical dataset

In conclusion, our study demonstrates that in terms of clustering structure and the stability of the technique under the parameters and data variations, t-SNE performs best on the skip-gram processed medical dataset compared to other four DR techniques. Although the analogy tests in the study do not show syntactic and semantic similarity, the results of this exploratory experiment can be used as initial knowledge for the more in-depth studies on word2vec in the medical field. There are a variety of semantic and syntactic questions that have been created. Because of time limitation, there are still some interesting analogy tests that we do not apply, which can be studies in the future. Although we know that the accuracy of word vectors increases after pre-process, we do not have the right to access machine learning in medical dataset. In the future work, we can pre-process medical text (e.g. removing all punctuation signs and transforming all words to lower-case) before training the data.

6 Bibliography

ACCUMULATE (2016). “Acquiring crucial medical information using language technology*”. IWT - Flemish Agency for Innovation, Science and Technology*.

Baud, R.H., A.M. Rassinous and J.R. Scherrer (1992). “Natural language processing and semantical representation of medical texts”. *Methods of Information in Medicine*, 31(2): 117-125.

Belkin, M. and P. Niyogi (2002). “Laplacian Eigenmaps and spectral techniques for embedding and clustering”. *In Advances in Neural Information Processing Systems*, 14: 585-591.

Bellman, R. (1961). “Adaptive Control Processes: A Guided Tour”. *Princeton University Press*, Princeton, New Jersey.

Borgognone, M.G., J. Bussi and G. Hough (2001). “Principal component analysis in sensory analysis: covariance or correlation matrix?”. *Food Quality and Preference*, 12(5): 323-326.

Brants, T., A.C. Popat, P. Xu, F.J. Och and J. Dean (2007). “Large language models in machine translation”. *In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning*, 2007: 858-867.

Brian, E. and H. Torsten (2011). “An introduction to applied multivariate analysis with R”. *Springer New York*: Chapter 3.

Brown, P.F., P. deSouza, R. Mercer and et al. (1992). “Class-based n-gram models of natural language”. *Computational Linguistics*, 18: 467-479.

Brun, A., H.J. Park, H. Knutsson and C.F. Westin (2003). “Coloring of DT-MRI fiber traces using Laplacian Eigenmaps”. *In Proceedings of the Eurocast 2003, Neuro Image Workshop*.

Carcía-Fernández, F.J., M. Verleysen, J.A. Lee and I. Díaz, (2013). “Sensitivity to parameter and data variations in dimensionality reduction techniques”. *Computational Intelligence and Machine Learning*, Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010 [Accessed: 17.06.2016].

Chernoff, H. (1973). “The use of faces to represent points in k-dmensional space graphically”. *Journal of the American Statistical Association*, 68: 361-368.

Choi, Y., C.Y. Chiu and D. Sontag (2016). “Learning Low-Dimensional Representations of Medical Concepts”. *Proceedings of the AMIA Summit on Clinical Research Informatics (CRI).*

Chowdhury, G.G. (2003), “Natural language processing”. *Annual Review of information Science and Technology*, 37(1): 51-89.

Cunningham, J.P. and Z. Ghahramani (2015). “Linear Dimensionality Reduction: Survey, Insights, and Generalizations”. *Journal of Machine Learning Research,* 16: 2859-2900.

Dryden, I.L. and K.V. Mardia (1998). “Statistical Shape Anlysis”. John Wiley & Sons.

Du, Q.S., Z.Q. Jiang and et al. (2006). “Amino Acid Principal Component Analysis (AAPCA) and its Applications in Protein Structural Class Prediction*”. Journal of Biomolecular Structure and Dynamics*, 23(6): 635-640.

Dumais, S. T., G. W. Furnas, T. K. Landauer, S. Deerwester and R. Harshman (1988). “Using latent semantic analysis to improve access to textual information”. *SIGCH Conference on Human Factors in Computing Systems*: 281-285.

Duraiswami, R. and VC. Raykar (2005). “The manifolds of spatial hearing”. *In Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 3: 285-288.

Elbatta, M.T.H and W.M. Ashour (2013). “A Dynamic Method for Discovering Density Varied Clusters”. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 6(1): 123-134.

Fang, Y.K., Z.Q., Huang and et al. (2014). “Research on Improve DBSCAN Algorithm Based On Ant Clustering”. *The Open Automation and Control System Journal*, 6: 1076-1084.

Fodor, I.K. (2002). “a survey of dimension reduction techniques”. *E-reports-ext.llnl.gov*.

Ghodsi Ali (2006). “Dimensionality reduction: a short tutorial”. Available from http://www.stat.washington.edu/courses/stat539/spring14/Resources/tutorial_nonlin-dim-red.pdf.

Hinton, G.E. and R.R. Salakhutdinov (2002). “Stochastic Neighbor Embedding”. *In Advances in Neural Information Processing Systems*, 15: 833-840.

Karami A. and R. Johansson (2014). “Choosing DBSCAN Parameters Automatically using Differential Evolution”. *International Journal of Computer Applications*, 91(7): 1-11.

Keim, D.A. (2000). “Designing pixel-oriented visualization techniques: Theory and applications”. *IEEE Transactions on Visualization and Computer Graphics*, 6(1): 59-78.

Kendall, D.G. (1989). “A survey of the Statistical Theory of Shape”. *Statistical Science*, 4(2): 87-99.

Kumar, N., A. Bansal and et al. (2014). “Chemometrics tools used in analytical chemistry: An overview”. Talanta, 123, 186-199.

Lee, J.A. and M. Verleysen (2005). “Nonlinear dimensionality reduction of data manifolds with essential loops”. *Neurocomputing*, 67: 29-53.

Lim, I.S., P.H. Ciechomski, S. Sarni and D. Thalmann (2003). “Planar arrangement of high-dimensional biomedical data sets by Isomap coordinates”. *In Proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems*: 50-55.

Lomonaco, V (2015). “Word2vec on the Italian language: first experiments”. Available from http://www.slideshare.net/VincenzoLomonaco/word2vec-on-the-italian-language-first-experiments.

Meystre, S. (2006). “Natural language processing to extract medical problems from electronic clinical documents: Performance evaluation”. *Journal of Biomedical Informatics*, 39(6): 589-599.

Mikolov, T., I. Sutskever, K. Chen and et al. (2013b). “Distributed representations of words and phrases and their compositionality”. *ArXiv preprint arXiv*: 1301.4546.

Mikolov, T., K. Chen, G. Corrado and J. Dean (2013a). “Efficient estimation of word representations in vector space”. *ArXiv preprint arXiv*: 1301.3781.

Mikolov, T., Wen-Tau Yih and G. Zweig (2013c). “Linguistic regularities in continuous space word representations”. *In HLT-NAACL*: 746-751.

Miñarro-Giménez, J.A., O. Marin-Alonso and M. Samwald (2015). “Applying deep learning techniques on medical corpora from the World Wide Web: a prototypical system and evaluation”. *arXiv*:1502.03682.

Mishra B.K., P. Sanurabh and B. Verma (2011). “A Novel Approach to Classify High Dimensional Datasets Using Supervised Manifold Learning”. 4th International Conference, ObCom 2011, ellore, TN, India, Part II. Proceedings: 22-30.

Muneeb, T.H., SK. Sahu and A. Anand (2015). “Evaluating distributed word representations for capturing semantics of biomedical concepts”. *Proceddings of the 2015 Workshop on Biomedical Natural Language Processing (BioNLP 2015)*: 158-163.

Niskanen, M. and O. Silvén (2003). “Comparison of dimensionality reduction methods for wood surface inspection”. *In Proceedings of the 6th international Conference on Quality Control by Artificial Vision*: 178-188.

Partridge, M. and R. Calvo (1997). “Fast dimensionality reduction and Simple PCA”. *Intelligent Data Analysis*, 2(3): 292-298.

Pereira, F., N. Tishby and L. Lee (1993).” Distributional clustering of English words”. *ACL*: 183-190.

Qu, L., G. Ferraro and et al. (2015). “Big data small data, in domain out-of domain, known word unknown word: the impact of word representations on sequence labelling tasks”. *Proceedings of the 19th Conference on Computational Language Learning*: 83-93.

Rai, P (2011). Nonlinear Dimensionality Reduction. CS5350/6350: *Machine Learning*. Available from https://www.cs.utah.edu/~piyush/teaching/25-10-slides.pdf.

Richard, S., C. Francois and M. Rohit (2016a). “Deep Learning for NLP: lecture notes Part I”. Available from http://cs224d.stanford.edu/lecture_notes/LectureNotes1.pdf.

Riitters, K.H., R.V. O’Neill and et al. (1995). “A factor analysis of landscape patter and structure metrics”. *Landscape Ecology*, 10(1):23-39.

Roweis, S.T. and L.K. Saul (2000). “Nonlinear dimensionality reduction by Locally Linear Embedding”. *Science*, 290(5500), 2323-26.

Rumelhart, D.E., G.E. Hintont and RJ. Williams (1986). “Learning representations by backpropagating errors”. *Nature*, 323(6088):533–536.

Saul, L.K. (2001). “An Introduction to Locally Linear Embedding”. Available from https://www.cs.nyu.edu/~roweis/lle/papers/lleintro.pdf.

Saul, L.K. and S.T. Roweis (2003). “Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds”. Journal of Machine Learning Research, 4: 119-155.

Saxena, A., A. Gupta and A. Mukerjee (2004). “Non-linear Dimensionality Reduction by Locally Linear Isomaps”. *Lecture Notes in Computer Science*, 3316:1038–1043.

Schnabel, T., I. Labutov, D. Mimno and T. Joachims (2015). “Evaluation methods for unsupervised word embeddings”. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing:* 298-307.

Scholz, M. (2006). “Approach to analyse and interpret biological profile data”. Ph.D. thesis.

Stegmann, MB. and DD. Gomez (2002). “A Brief Introduction to Statistical Shape Analysis”.

Steinbach, M., L. Ertoz and V. Kumar (2003). “The Challenges of Clustering High Dimensional Data”. In L. T. Wille, editor, *New Vistas in Statistical Physics – Applications in Econophysics, Bioinformatics, and Pattern Recognition*. Springer-Verlag.

Strange, H. and R. Zwiggelaar (2011). “A Generalised Solution to the Out-of-Sample Extension Problem in Manifold Learning”. *In AAAI Conference on Artificial Intelligence.*

Tan, P.N., M. Steinbach and V. Kumar (2006). “Introduction to Data Mining”. Chapter 8, Cluster Anlysis: Basic Concepts and Algorithms.

Tenenbaum, J., D.S. Vin and L., John (2000). “A global geometric framework for nonlinear dimensionality reduction”. *Science*, 290(5500): 2319-23.

TensorFlow (2016). “Vector Representations of Words”. Available from https://www.tensorflow.org/versions/r0.10/tutorials/word2vec/index.html.

Torgerson, W.S. (1952). “Multidimensional scaling I: Theory and method”. *Psychometrika*, 17: 401-419.

Trikha, P and S. Vijendra (2013). “Fast Density Based Clustering Algorithm”. *International Journal of Machine Learning and Computing*, 3(1): 10-12.

Turian, J., L. Ratinov and Y. Bengio (2010). “Word representations: a simple and general method for semi-supervised learning”. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*: 384-394.

Van de Maaten, L. (2013). “Barnes-Hut-SNE*”. In Proceedings of the International Conference on Learning Representations*.

Van der Maaten, L. (2009). “Learning a Parametric Embedding by Preserving Local Structure”. *In Proceedings of the Twelfth International Conference on Artificial Intelligence & Statistics (AI-STATS), JMLR W&CP 5*:384-391.

Van der Maaten, L. and G. Hinton (2008a). “Visualizing data using t-SNE*”. Journal of Machine Learning Research*, 9: 2579-2605.

Van der Maaten, L., E. Postma and J. Van den Herik (2008b). “Dimensionality reduction: a comparative review”. Online Preprint.

Wang, J.Z (2011). “Geometric Structure of High-Dimensional Data and Dimensionality Reduction”. Chapter 10.

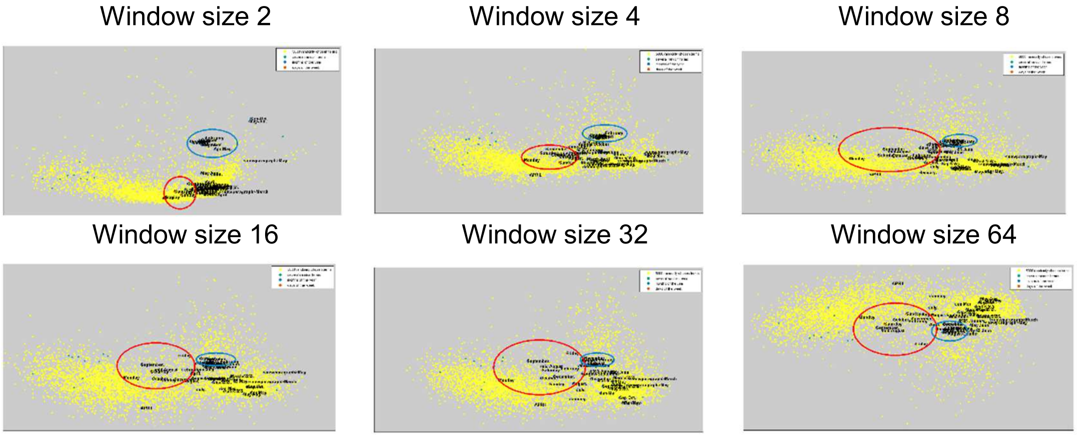
Wolfgang Karl, H. and L. Simar (2012). “Applied Multivariate Statistical Analysis”. *Springer New York*: Chapter 10.

Zhai, M, J. Tan and J.D. Choi (2015). “Intrinsic and Extrinsic Evaluations of Word Embeddings”. *In Thirtieth AAAI Conference on Artificial Intelligence*.

Appendix A

1. A 5000 randomly-chosen-words dataset

Figure 29: PCA on 5000 randomly chosen-words datasets with a different window size


Window size 2

Window size 4

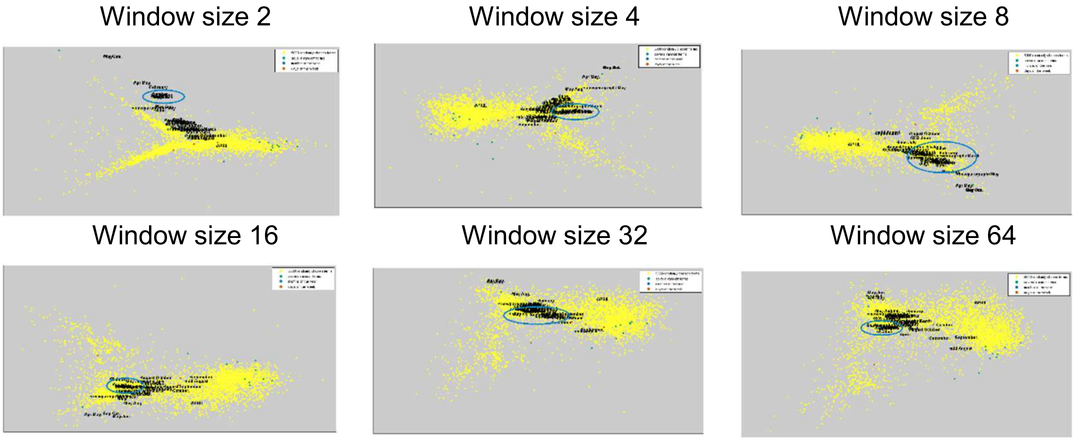
Window size 8

Window size 16

Window size 32

Window size 64

Figure 30: Isomap on 5000 randomly chosen-words datasets with a different window size


Window size 2

Window size 4

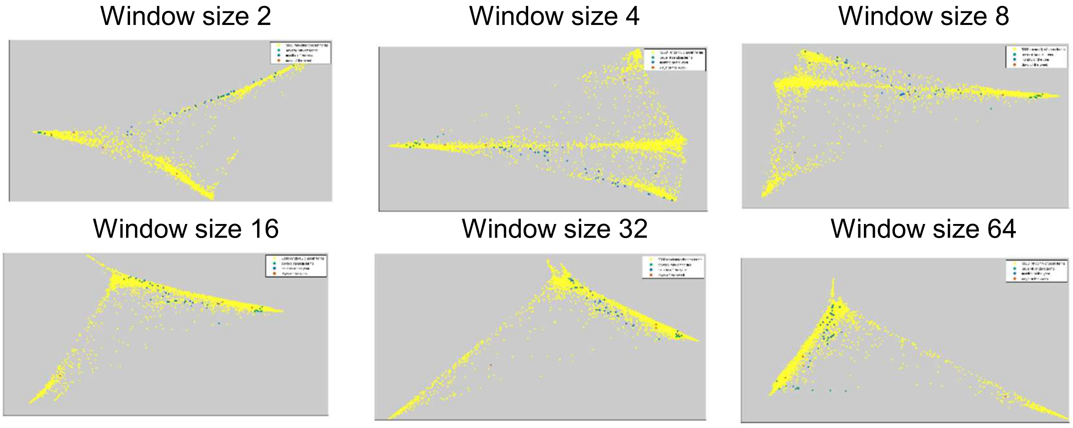
Window size 8

Window size 16

Window size 32

Window size 64

Figure 31: LE on 5000 randomly chosen-words datasets with a different window size


Window size 2

Window size 4

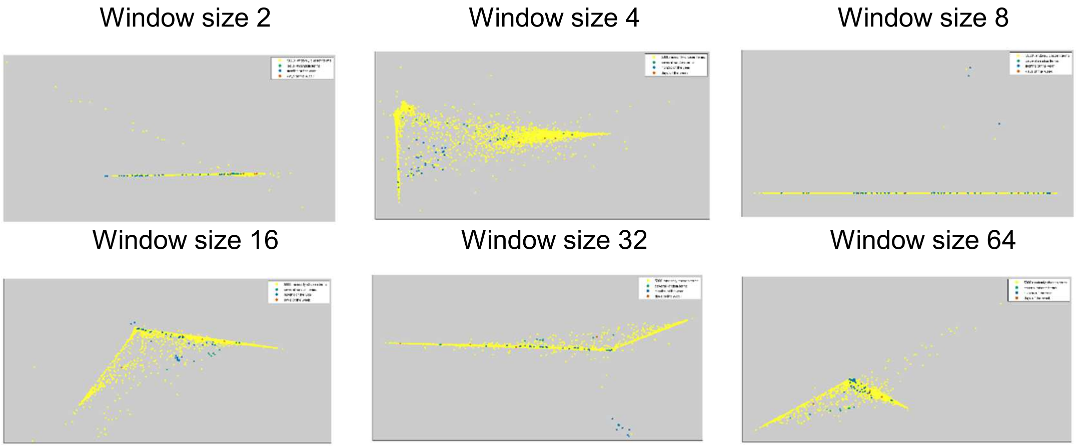
Window size 8

Window size 16

Window size 32

Window size 64

Figure 32: LLE on 5000 randomly chosen-words datasets with a different window size


Window size 2

Window size 4

Window size 8

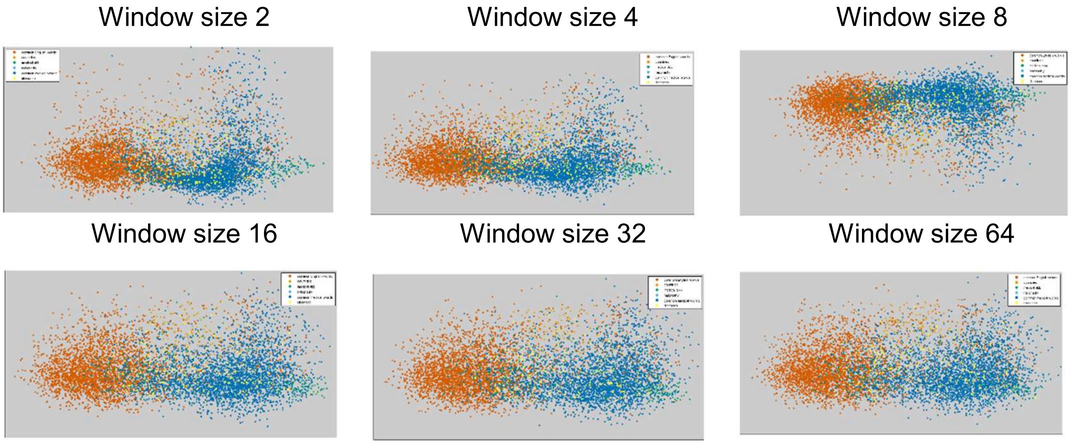
Window size 16

Window size 32

Window size 64

2. A six ‘word lists’ filtered dataset

Figure 33: PCA on the six ‘word lists’ filtered dataset with a different window size


Window size 2

Window size 4

Window size 8

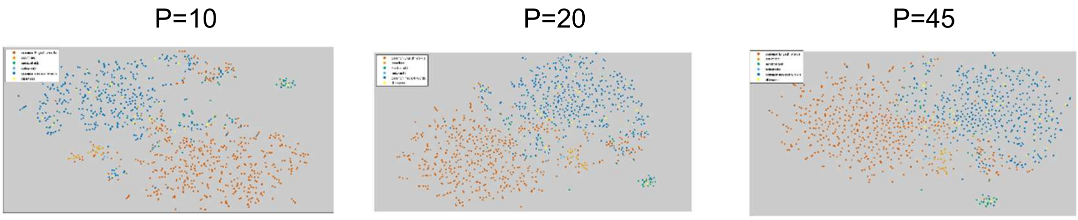
Window size 16

Window size 32

Window size 64

3. The stability of the DR methods

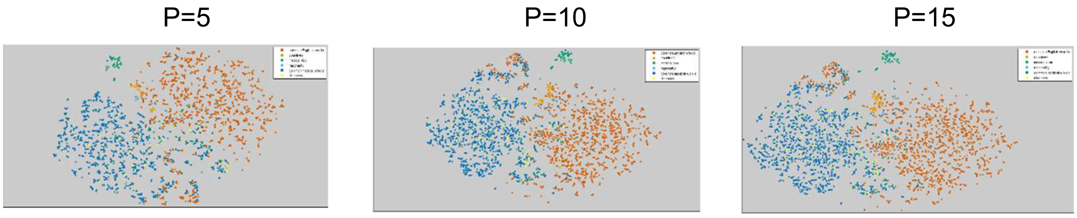
Figure 34: t-SNE on 1000-term dataset


P=10

P=20

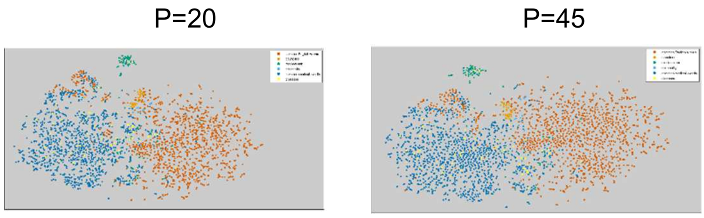
P=45

Figure 35: t-SNE on 2500-term dataset


P=5

P=10

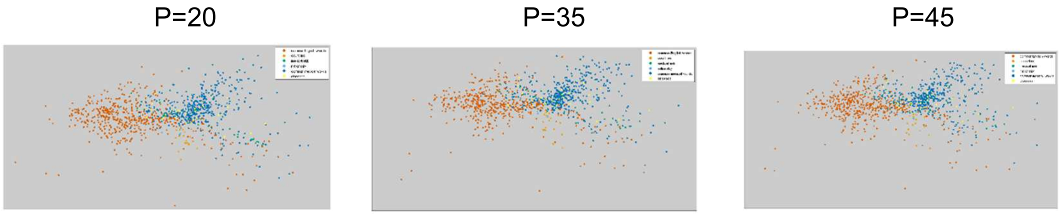
P=15



P=20

P=45

Figure 36: Isomap on 1000-term dataset

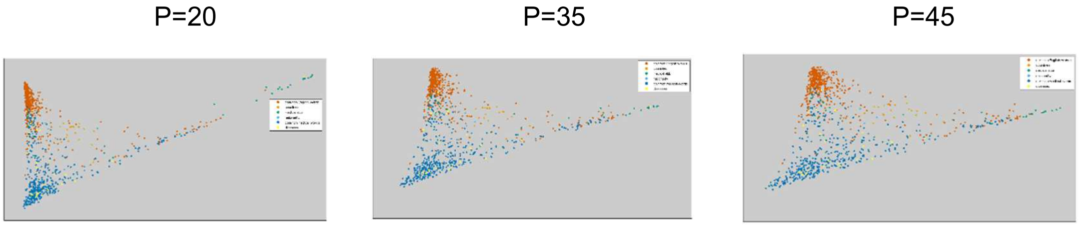


P=20

P=35

P=45

Figure 37: LE on 1000-term dataset

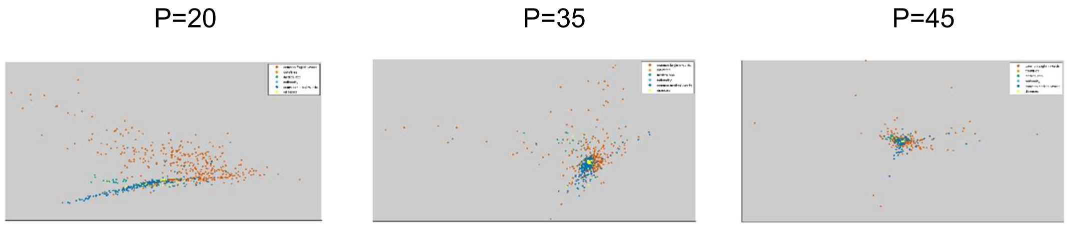


P=20

P=35

P=45

Figure 38: LLE on 1000-term dataset



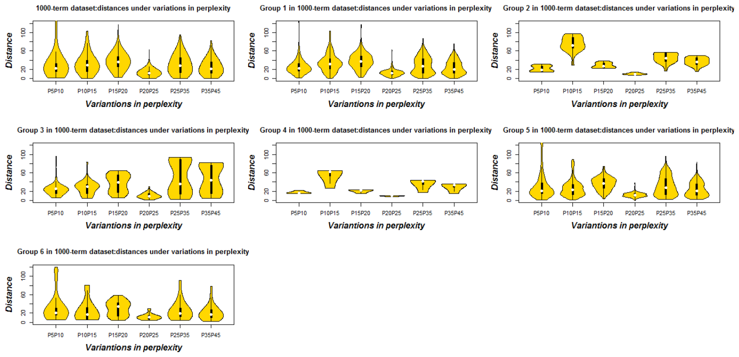
P=20

P=35

P=45

Figure 39: Violin plots in 1000-, 1500- and 2000-term dataset

d=1000



1000-term dataset:distances under variations in perplexity

Group 1 in 1000-term dataset:distances under variations in perplexity

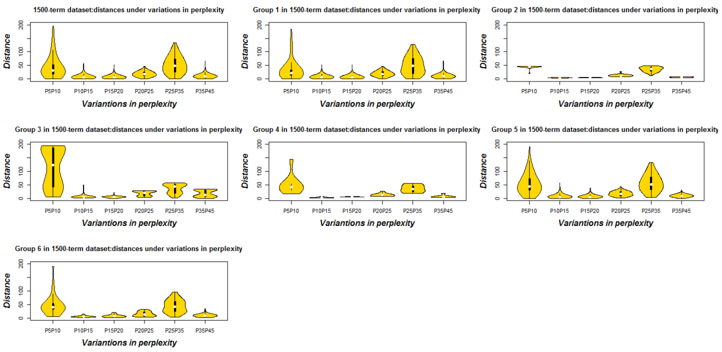
Group 2 in 1000-term dataset:distances under variations in perplexity

Group 3 in 1000-term dataset:distances under variations in perplexity

Group 4 in 1000-term dataset:distances under variations in perplexity

Group 5 in 1000-term dataset:distances under variations in perplexity

Group 6 in 1000-term dataset:distances under variations in perplexity d=1500


1500-term dataset:distances under variations in perplexity

Group 1 in 1500-term dataset:distances under variations in perplexity

Group 2 in 1500-term dataset:distances under variations in perplexity

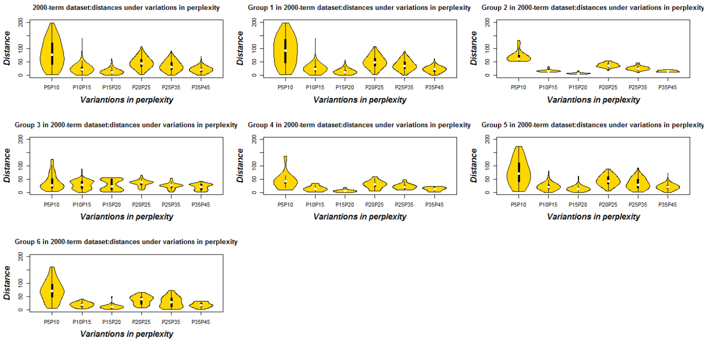
Group 3 in 1500-term dataset:distances under variations in perplexity

Group 4 in 1500-term dataset:distances under variations in perplexity

Group 5 in 1500-term dataset:distances under variations in perplexity

Group 6 in 1500-term dataset:distances under variations in perplexity

d=2000


2000-term dataset:distances under variations in perplexity

Group 1 in 2000-term dataset:distances under variations in perplexity

Group 2 in 2000-term dataset:distances under variations in perplexity

Group 3 in 2000-term dataset:distances under variations in perplexity

Group 4 in 2000-term dataset:distances under variations in perplexity

Group 5 in 2000-term dataset:distances under variations in perplexity

Group 6 in 2000-term dataset:distances under variations in perplexity

Table 8: The relative distance test in Past tense

Verbs	w2 vs w1	w4 vs w1	w8 vs w1	w16 vs w1	w32 vs w1	w64 vs w1
admit - admitted	0.26	-0.73	-0.85	-0.92	-1.02	-0.03
advocate - advocated	-0.35	-0.38	-0.35	0.83	-0.33	0.75
aid - aided	-0.37	-0.94	-0.95	-0.77	-0.90	-0.93
assess - assessed	-0.48	-1.21	-1.43	-1.31	-1.26	-1.28
assign - assigned	-0.28	0.26	-0.32	0.97	0.52	0.35
assist - assisted	0.24	-0.02	0.29	-0.37	-0.68	-0.45
care - cared	-0.50	-0.08	-0.33	-0.58	-0.40	-0.63
charge - charged	-0.03	-0.05	-0.06	-0.07	-0.10	-0.10
counsel - counseled	-1.06	-1.05	-1.04	-1.08	-1.12	-1.12
cure - cured	-1.09	0.18	-1.12	-1.13	-1.17	-1.16
diagnose - diagnosed	-0.02	-0.93	-0.44	-1.14	-1.43	-0.95
evaluate - evaluated	-0.52	-1.08	-1.01	-1.17	-1.20	-1.05
heal - healed	-0.94	-0.96	-0.97	-0.96	-0.97	-0.97
help - helped	-0.52	0.34	-0.27	-0.42	0.05	0.08
influence - influenced	-0.73	-0.75	-0.73	-0.77	-0.78	-0.80
monitor - monitored	-0.59	-0.93	-1.51	-0.66	-1.30	-1.02
practice - practiced	0.10	-0.04	-0.04	0.18	-0.11	0.44
prevent - prevented	-0.28	-0.21	0.07	-0.91	-0.93	-0.96
proceed - proceeded	-0.01	-0.01	0.34	0.09	0.28	0.55
provide - provided	-1.09	-1.15	-1.23	-1.24	-0.22	-0.09
qualify - qualified	0.05	-0.02	0.18	-0.01	-0.02	0.00
refer - referred	0.66	-0.01	0.75	0.97	-0.02	-0.05

regulate - regulated	-0.29	-0.34	-0.35	-0.35	-0.34	-0.36
repair - repaired	-0.03	-0.06	-0.06	-0.07	-0.06	-0.07
restrict - restrict	-0.24	-0.27	-0.23	0.18	-0.23	0.23
support - supported	-0.10	0.04	0.29	-0.51	-0.94	-0.25
treat - treated	-0.02	0.10	-0.05	-0.69	-1.16	-0.51
the number of decreased "the relative distance"	22.00	22.00	21.00	21.00	24.00	21.00

Appendix B

The matlab codes of PCA, t-SNE, Isomap, LE and LLE are based on dimension reduction toolbox provided by Laurens van der Maaten, available from https://lvdmaaten.github.io/drtoolbox/. Because of space limitation, we will only represent one example of the similar codes. For example, we will only show the codes for the window size 1 dataset since the codes for the other window-sized datasets are the same. '%' and ' #' represent matlab and R codes, respectively.

1. Dimensionality reduction methods

1.1. Data selection

```
# read the original data
data1 <- read.delim("C:/Users/Administrator/Desktop/thesis/pca and t-SNE comparision/process original data/d50 w1txt.txt", header=F)
# select 5000 terms from the data
set.seed(123)
sub1 <- data1[sample(147764, 5000),]
# select 3-category word list
sub11 <- data1[c(3,4,5,6,12,15,16,19,20,21,22,1547,1646,2967,3268,3295,3300,3410,3705,3852,

    3895,3918,15974,24872,41878,47810,49997,53128,53437,65256,69793,71606,75880,

    77808,80507,87910,87983,88050,89032,89948,94732,98688,99915,104011,105549,

    107366,107871,108043,108921,112728,115111,118886,125276,127705,135290,140922),]
# combine 5000 with sub11
sub12 <- rbind(sub1, sub11)
sub22 <- sub12[!duplicated(sub12),]
# export 5000 randomly chosen-words data
library(xlsx)
write.xlsx(sub22, "c:/data5001.xlsx")

# six 'word lists' filtered dataset
data1 <- read.delim(choose.files(), header=F) # the original window size 1 dataset
data2 <- read.delim(choose.files(), header=F) # 6000 common words
data3 <- read.delim(choose.files(), header=F) # countries and continents
data4 <- read.delim(choose.files(), header=F) # medical abbreviation
```

```
data5 <- read.delim(choose.files(), header=F) # people by countries
data6 <- read.delim(choose.files(), header=F) # medical
data7 <- read.delim(choose.files(), header=F) # disease
df1 <- data1[data1$V1 %in% data2$V1 & data1$V1 %in% data3$V1& data1$V1
%in% data4$V1& data1$V1 %in% data5$V1& data1$V1 %in% data6$V1&
data1$V1 %in% data7$V1,]
df1 <- df1[!duplicated(df1),]
write.table(df1, file="df1.txt", sep=" ", row.names = F)
```

1.2. 5000 randomly-chosen-words and six 'word lists' dataset

```
a = matrix_5001;
b = matrix_5001name;
no_dims = 2;
% PCA
[mappedXpca, mappingpca] = pca(a, no_dims);
% Isomap
k=12;
[mappedXi12, mappingi12] = isomap(a, no_dims, k);
% LE
eig_impl = 'Matlab';
sigma = 1;
k=12;
[mappedXl12, mappingl12] = laplacian_eigen(a, no_dims, k, sigma,
eig_impl);
% LLE
k=12;
mappedXlle12 = lle(a, no_dims, k, eig_impl);
% tsne
% perplexity = 30;
perplexity = 30;
initial_dims = 50;
mappedX30=tsne(a, [], no_dims, initial_dims, perplexity);

% PCA plot in 5000 randomly chosen-words dataset
dpg1 = mappedXpca(mappedXpca(:,3)==1, :);
dpg2 = mappedXpca(mappedXpca(:,3)==2, :);
dpg3 = mappedXpca(mappedXpca(:,3)==3, :);
dpg4 = mappedXpca(mappedXpca(:,3)==4, :);
figure(1);
scatter(dpg1(:,1), dpg1(:,2), 15, 'MarkerEdgeColor',[255 255
51]./255, 'MarkerFaceColor', [255 255 51]./255);hold on
scatter(dpg2(:,1), dpg2(:,2), 15, 'MarkerEdgeColor',[0 158 115]./255,
'MarkerFaceColor', [0 158 115]./255);
scatter(dpg3(:,1), dpg3(:,2), 15, 'MarkerEdgeColor',[0 114 178]./255,
'MarkerFaceColor', [0 114 178]./255);
scatter(dpg4(:,1), dpg4(:,2), 15, 'MarkerEdgeColor',[213 94 0]./255,
'MarkerFaceColor',[213 94 0]./255);
legend({'5000 randomly chosen terms', 'several random terms', 'months
of the year', 'days of the week'}, 'FontSize',8);
set(gca,'Color',[0.8 0.8 0.8]);
set(gca,'YTick',[]);
set(gca,'XTick',[]);

# the optimal value in DBSCAN
dev.new()
kNNdist(d5001, k=10, search="kd")
```

```
kNNdistplot(d5001, k=10)
% DBSCAN plot
figure(1);
epsilon=10;
MinPts=4.5;
IDX1=DBSCAN(mappedX30,epsilon,MinPts);
PlotClusterinResult(mappedX30, IDX1);
```

1.3 The stability of dimensionality reduction methods

```
% PCA
no_dims = 2;
[mappedXpca, mappingpca] = pca(matrix_1000, no_dims);
% Isomap
mappedXi7 = zeros(1000, 14);
a=1;
b=2;
for i=5:5:45;
[mappedXi, mappingi] = isomap(matrix_1000, no_dims, i);
mappedXi7(:, a:b) = mappedXi;
a = a+2;
b = b+2;
end
% LE
eig_impl = 'Matlab';
sigma = 1;
a=1;
b=2;
mappedXl7 = zeros(1000, 14);
for i=5:5:45;
[mappedXl, mappingl] = laplacian_eigen(matrix_1000, no_dims, i, sigma,
eig_impl);
mappedXl7(:, a:b) = mappedXl;
a = a+2;
b = b+2;
end
% LLE
a=1;
b=2;
mappedXlle7 = zeros(1000, 14);
for i=5:5:45;
[mappedXlle, mappinglle] = lle(matrix_1000, no_dims, i, eig_impl);
mappedXlle7(:, a:b) = mappedXlle;
a = a+2;
b = b+2;
end
% tsne
initial_dims = 50;
a=1;
b=2;
mappedX7 = zeros(1000, 14);
for p=5:5:45;
[mappedX7, mapping7] = tsne(matrix_1000, [], no_dims, initial_dims,
p);
mappedX7(:, a:b) = mappedX7;
a = a+2;
b = b+2;
```

```
end
% Procrustes analysis of isomap on 1000-term dataset with k 5 and 10
X1 = mappedXil05(1:1000,1:2);
X2 = mappedXil010(:,1:2);
[d1,Z1,tr1] = procrustes(X1,X2);

# Clustering information analysis in t-SNE
# d1000
data1000 <-
read.delim("C:/Users/Administrator/Desktop/d1000distancetxt.txt",
header=F)
# summary
colMeans(data1000[,-1])
sapply(data1000[,-1], sd)
# Violin plots
library(vioplot)
dev.new()
x1 <- data1000$V2
x2 <- data1000$V3
x3 <- data1000$V4
x4 <- data1000$V5
x5 <- data1000$V6
x6 <- data1000$V7
vioplot(x1, x2, x3, x4, x5, x6,
names=c("P5P10","P10P15","P15P20","P20P25","P25P35","P35P45"),
col="gold", ylim =c(0,120))
title("Group 1 in 1000-term dataset:distances under variations in
perplexity")
mtext("Variantions in perplexity", side = 1, line = 3, cex = 1, font
= 4)
mtext("Distance", side = 2, line = 3, cex = 1, font = 4)

parallel coordinates
labels = {'P5P10','P10P15','P15P20','P20P25','P25P35','P35P45'};
figure(1);
parallelcoords(d1000g1,'Group',d1000g1_label,'Labels',labels);
grid on;
```

1.4 Word vector evaluation

Here we present codes for nouns test since codes for other test are similar.

```
% calculate tsne on seven datasets
no_dims=2;
initial_dims=50;
perplexity=30;
mappedXn_d1 = tsne(nvector_d1, [], no_dims, initial_dims, perplexity);
mappedXn_d2 = tsne(nvector_d2, [], no_dims, initial_dims, perplexity);
mappedXn_d3 = tsne(nvector_d3, [], no_dims, initial_dims, perplexity);
mappedXn_d4 = tsne(nvector_d4, [], no_dims, initial_dims, perplexity);
mappedXn_d5 = tsne(nvector_d5, [], no_dims, initial_dims, perplexity);
mappedXn_d6 = tsne(nvector_d6, [], no_dims, initial_dims, perplexity);
mappedXn_d7 = tsne(nvector_d7, [], no_dims, initial_dims, perplexity);
% plot window size 1 dataset
figure(1);
gscatter(mappedXn_d1(:,1), mappedXn_d1(:,2), nvector_d1label, 'br');
text(mappedXn_d1(:,1), mappedXn_d1(:,2), nvector_d1name);
```

```
legend({'Singular nouns', 'Plural nouns'},
'FontSize',10,'FontWeight','bold');
title('Plural nouns test on the dataset with window size of 1',
'FontSize', 14, 'FontWeight','bold');
x=mappedXn_d1(:,1);
y=mappedXn_d1(:,2);
hold on
for i=1:2:75
line([x(i) x(i+1)], [y(i) y(i+1)], 'Color', 'k', 'LineStyle','--')
end

% calculate “the relative distance”
a=1;
 d=zeros(76,266);
for j=1:2:76
    for i=1:76
    x1=[mappedXn_d1(j,1) mappedXn_d1(j,2)];
    y1=[mappedXn_d1(i,1) mappedXn_d1(i,2)];
    d(i,a)=pdist2(x1,y1, 'euclidean');
    end
a=a+1;

    for i=1:76
    x1=[mappedXn_d2(j,1) mappedXn_d2(j,2)];
    y1=[mappedXn_d2(i,1) mappedXn_d2(i,2)];
    d(i,a)=pdist2(x1,y1, 'euclidean');
    end
    a=a+1;

    for i=1:76
    x1=[mappedXn_d3(j,1) mappedXn_d3(j,2)];
    y1=[mappedXn_d3(i,1) mappedXn_d3(i,2)];
    d(i,a)=pdist2(x1,y1, 'euclidean');
    end
    a=a+1;

    for i=1:76
    x1=[mappedXn_d4(j,1) mappedXn_d4(j,2)];
    y1=[mappedXn_d4(i,1) mappedXn_d4(i,2)];
    d(i,a)=pdist2(x1,y1, 'euclidean');
    end
    a=a+1;

    for i=1:76
    x1=[mappedXn_d5(j,1) mappedXn_d5(j,2)];
    y1=[mappedXn_d5(i,1) mappedXn_d5(i,2)];
    d(i,a)=pdist2(x1,y1, 'euclidean');
    end
    a=a+1;


    for i=1:76
    x1=[mappedXn_d6(j,1) mappedXn_d6(j,2)];
    y1=[mappedXn_d6(i,1) mappedXn_d6(i,2)];
    d(i,a)=pdist2(x1,y1, 'euclidean');
    end
    a=a+1;

    for i=1:76
    x1=[mappedXn_d7(j,1) mappedXn_d7(j,2)];
    y1=[mappedXn_d7(i,1) mappedXn_d7(i,2)];
    d(i,a)=pdist2(x1,y1, 'euclidean');
    end
    a=a+1;

end
```

AFDELING
Straat nr bus 0000
3000 LEUVEN, BELGIË
tel. + 32 16 00 00 00
fax + 32 16 00 00 00
www.kuleuven.be