Contents

List of Tables

List of Figures

Chapter 1

Introduction

The increasing numbers of textual documents from diverse sources such as different websites (social networks, news, magazines, blogs and medical recommendation websites), publications and articles and medical prescriptions leads to massive amounts of daily complex data. This phenomenon has caused many researchers to focus on analysing the content and measuring the similarities among the documents and texts to cluster them. One of challenges is to measure the similarity among texts in order to cluster similar documents and distinguish texts based on subjects discussed and covered in each text or document. Measuring similarities among documents also helps to identify which documents are more or less similar to a particular topic or document of interest, which is widely used in recommendation systems. The desired output of a recommendation system is a ranked top n items to a particular item or searched keyword.

There are several algorithms used for measuring similarity between documents such as Pearson (Kornbrot 2005), Spearman (Zar 1998), Kullback-Leibler divergence (Kullback & Leibler 1951), Jaccard coefficient (Jaccard 1912), Shannon (Wartena & Brussee 2008), Euclidean Distance (ED) and Cosine similarity (Salton & Buckley 1988). One popular method to measure the similarity between documents is to represent the terms within the documents as vectors and measure the similarity among them based on the angle or Euclidean distance between each pair. By only considering these two criteria for similarity measurement, we may miss important underlying semantic similarities in this area. We propose a new method, TS-SS, to measure the similarity level among documents, in such a way that one hopes to better understand which documents are more (or less) similar. Though the results of many studies (Salton & Buckley 1988, Nelson et al. 2004) show the geometric and VSM based models are more robust in measuring similarities among documents compared to non-geometric models, there are not many works focusing on geometric methods to boost these similarity measures for better results. That is why in this study we focus on geometric methods.

The main purpose of this study is to measure the similarities among documents with high accuracy in order to have a better understanding of which documents are more similar (or less similar). We call this concept *similarity level* which is focused in this study. In this work, *accuracy* refers to the power of a measure in differentiating the similarity level among documents in such a way that one can understand which documents are less, more and most similar. This power of differentiation can be significantly useful for recommendation systems and clusterings.

As mentioned earlier, this study focuses on geometric similarity measures which are popular for document clustering but there are not much works on improvement of the current geometric models in such a way that can be used for measuring a concept called *similarity level*. In some research and surveys (Nelson et al. 2004, Salton & Buckley 1988), diverse similarity measures used for IR have been evaluated and results show Cosine similarity outperforms other measures. The applicability of the similarity level will be explain more in details in next sections. This study gives insights on the drawbacks of geometrical and some non-geometrical similarity measures and provides a novel method to combine the other geometric criteria into a method to measure the similarity level among documents from new prospective.

This study contains the following chapters: in Chapter 2 related studies and research to recommendation systems, vector space models, document clustering and combination of these tree categories are explained briefly. In Chapter 3 text preprocessing techniques to prepare the documents are described. Secondly we explain how the Vector Space Model forms vectors from body of texts. Then we explain how traditional geometric similarity measures namely Cosine similarity and Euclidean distance use vectors associated to each document to identify the similar and dissimilar documents/texts. Finally, clustering techniques, clustering tools and methods for evaluating clusters are explained in details. In Chapter 4 drawbacks of Cosine similarity and Euclidean distance have been scrutinized from different views and it is explained why existing geometric measures are not robust enough to measure the similarity level accurately. In Chapter 5 a new method called TS-SS proposed, which covers the mentioned drawbacks to measure the similarity level among documents. In Chapter 6 five datasets used for experiments and four evaluation methods for making comparison among the measures are described in detail. In Chapter 7 the proposed method and other similarity measures are applied on five datasets to cluster the documents and then and the results of clustering from the proposed method and other similarity measures are compared using four evaluation methods. The evaluations' results show the proposed model, TS-SS outperforms the other measures. Finally in Chapter 8 the time complexity of Cosine similarity, Euclidean distance, TS-SS and K-Means algorithm which is used for clustering are calculated.

Chapter 2

Related Work

In this section we briefly mention some works which have used diverse similarity measures and methods for clustering similar objects or design a recommendation systems among users in social networks, among textual documents in documents datasets or among webpages. All the works use pairwise and distance similarity and majority of them calculate the similarities among objects based on textual contents which is exactly what we do. We also tried to cite some works which use Vector Space Model (VSM) which is the base of our work.

Liao *et al.* proposed a hybrid friend recommendation system for a Virtual World (VW) based on the similarity level and strength. In VW, users interact in an electronic environment which is mimics the physical space. Each user is represented as Avatar and the main concept of interaction among users is same as the social networks. The proposed hybrid recommendation system is built based on the user similarities and virtual contact strengths. The system is created in three main steps. At first, the pairwise similarity is calculated with respect to an attribute quantitatively and non-quantitatively. Let $d_i(u)$ and $d_i(v)$ be the respective value of attribute i for user u and user v. Then the value of non-quantitative similarity value of 1 indicates u and v are similar in terms of attribute i, and 0 means they are not similar. The quantitative similarity is computed based on the distance between $d_i(u)$ and $d_i(v)$. This similarity can be represent as $1 - \left|d_i(u) - d_i(v)\right|$. In terms of attribute i, if u and v are more similar than x and y, then the distance between u and v is shorter than between x and y (similarity level). The same definition is used in our research for the similarity level concept. Then the similarity level is harmonized, the weighted mean distance is applied. In second phase, the interaction frequency between users is used to gain the contact strength. Finally, the similarity level and the contact strength are merged in SVM calculations and KNN classifier is used as predicting classifiers. Then the predicted values are used to rank the similarity levels between similar users. The method have been tested on portion of data collected from "www.roomi.com.tw" and the results are evaluated using recall-precision method and the comparison between the hybrid model and the classic KNN and SVM model show the better results (Liao et al. 2015).

Han *et al.* investigate the effect of social features such as demographic information on prediction of similarity or dissimilarity between each pair of users where the demographic information about one of the users in a pair is available and about the other one is limited. They used Facebook dataset which contains 479,048 users and 5,263,351 user-generated interests, and the homophily interest is calculated across three domains namely movies, music and TV shows. Analysing the data shows that it can be split in three parts: user interests (movies, music, TV, shows, books, games, athletes, teams, sports, and activities), Demographic information (age, gender, current city, home town, high school, college and employer) and social relationship. Users are divided in two categories of “Active users” (represented as u_a, those who present their demographic information) and “Passive users” (represented as u_p, those who only report partial demographic information and/or friendships, but hide interests from the public).The main goal is to determine u_a and u_p are similar or dissimilar in interest and select a subset of active users who probably share many interests with u_p, given a u_p and a set of active users. Interest similarities are computed by two methods namely binary similarity (if two users have any mutual interests, it is 1 and it is 0 if there is no interest in common) and Cosine similarity. Then using the measured similarities, the collective interest similarity over an aggregation of user pairs (C) is estimated. The pair set C is generated by considering two factors: (1) the related profile attribute and (2) the focused interest domain. Then SVM based with 10-cross validation predication model is used to label u and v as either interest-similar or interest-dissimilar by learning their social features. For an specified threshold, if interest similarity between u and v is more than the threshold, the users are similar, otherwise they are dissimilar (Han et al. 2015).

Nitai *et al.* proposed a friend recommendation system based on the graphical structure of social networks using the concept of user’s friends and friends-of-friends (FOF) to find the similar users. Their algorithm analyses each user and the related subgraph formed by their connected people separately by three degrees of separation, though only users separated by two degrees are taken into consideration for the friend suggestion purpose. The target social network for the study is a small network called Oro-Aro and the system is developed based on three indexes. The first index is defined as the number of adjacent nodes, the second index is calculated based on the density of the measured result from the first index, and the third index computes the density of the group formed by the adjacent vertices (Silva et al. 2010).

Xie measures the similarity among users from context (location, time) and content point of view to develop a friend recommendation system in social networks. First the rate of interest is computed per each blog using Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF model is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus and is explained in details in Section 3.2. Then the similarity based on the set of interest between each pair of users is calculated using Jaccard coefficient. Finally, the Cosine similarity is used to measure the similarity between each pair from another prospective. The combination of measures similarities is used to give assign a weight among users and consequently find the k nearest neighbors as the candidates for recommendation. The candidates are re-rated based on the geographical locations retrieved from their IP address. The results are evaluated by Precision-Recall metric. In average, the high precision is about 50% when the recall falls below 60% from the result of 8 studies (Xie 2010).

Hannon *et al.* studied the challenge of finding friends (followees) who are highly similar and pertinent to each other based on their interests. The system called Twittomender, mines 1 million Twitter users' profile by following the links between them. The system can find the high related and similar users for any desired search query. The system simply uses TF-IDF to score the users based on their similarity to the search query and returns top 20 similar users. For evaluation, 20,000 user documents are selected, 1000 picked as a user test set and the rest selected as recommendation pool. Then after recommending process, it is checked to see whether or not a suggested user was already being followed by that test user or not (Hannon et al. 2011). Balabanovic and Shoahm proposed a content-based and collaborative-based recommendation system which can be applied for recommending items from some fixed database. The content-based system recommends an item/web page to a user based on if the content/relevant text is similar to user's interest. The collaborative approach recommends favourite items from other similar users. The system called Fab gives high weight to discriminative words and recommends Web pages to users, by showing Web page contents with the 100 most important words (Balabanović & Shoham 1997).

Pazzani and Billsus designed Syskill & Weber to identify which web pages are interesting or uninteresting on a particular topic. The system finds top k informative words using a hybrid method similar to TF-IDF. Then they compare the information collected from profile of users including their interests, to the informative words collected from each page, and find the most similar web pages to each user (Pazzani et al. 1996).

Sahami and Heilman proposed a kernel function to measure the similarity between short texts. First x is issued as a query. Then for the retrieved documents using query x, TF-IDF is used to weight the words in each document, and finally the similar documents are identified based on the weights of terms (Sahami & Heilman 2006). One of the issues with this method is that, it is applicable for short texts only, because passing a long text as a query, is an expensive operation.

Bilenko and Mooney used TF-IDF and Cosine similarity approach to detect the duplications in database. In fact they measured the similarity among pairs of records in database, and detected the similar records if two specific field like address in two records are highly similar (Bilenko & Mooney 2003).

Mihalcea *etl al.* measured the similarity between two given texts based on the corpus and knowledge of the texts. They take the word specificity of words, as an indicator to assign a higher weight to a semantic matching identified between two specific words like collie and sheepdog), and assign less weight to the similarity measured between generic concepts like get and become. For this purpose again a hybrid TF-IDF is used. They also used PMI-IR method suggested by Turney (2001) (Turney 2001) for unsupervised evaluation of the semantic similarity of words from another prospective (Mihalcea et al. 2006).

Soucy and Mineau evaluate some hybrid TF-IDF models and use the main concept of the VSM to define their own weighting methods based on the confidence for document categorization. The model is similar to TF-IDF with an additional factor which is the number of categories. They also define a minimum and maximum confidence threshold to decide each document belongs to which category. Then finally they examine the accuracy of their method for different confidence thresholds to the traditional TF-IDF and the other hybrid versions explained in their literature, to find the best threshold for the datasets used in their study (Soucy & Mineau 2005).

Mao and Chu present a phrase-based VSM method to measure the similarity between by taking the similarity between phrases and their conceptual similarity and their common word stems. They represent each document as a set of phrases. Each phrase may correspond to n multiple concepts (due to polysemy). So they use TF-IDF to find out the closest concept category which each phrase might belong. Then among the related documents which are from the same concept, the similarity is measured using Cosine similarity to find out which documents are similar (Mao & Chu 2002).

Becker and Kuropka developed Topic-based Vector Space Model (TVSM) to measure the document similarity. In this approach, each term in the document is represented as vector space and the weight of the term is defined as calculated as the length of vector. In the paper an example is shown to illustrate how the direction and length of the vectors play a role in finding the important terms in the documents and weight assignments. Terms which have the longer magnitude (which meet the predefined length threshold) and tighter angle (close to 0), are considered as the important terms. Then after the weights of terms in each document is used to calculate the Cosine similarity between each pair of documents. The main concept of this work is similar to our study as they took the magnitude and direction of vectors into account as well (Becker & Kuropka 2003).

Wong *et al*. defined a generalization of the SVM, called the GSVM for resolving the limitations associated to Boolean retrieval related to some terms in document in term-based VSM. The stated problem for the study is “how Boolean algebra may be modelled as vectors in a vector space”. This paper mentions to the Boolean drawbacks of the VSM, which is explained and used as the motivation for our study as well (Wong et al. 1985).

Nelson *et al*. studied and compared two recommendation server methodologies implemented for the NASA Technical Report Server (NTRS). One method is the log analysis and the other one is VSM. They measured the similarities using Cosine similarity to recommend top 10 similar documents. After running the experiments they found out, in general, Cosine outperforms the log analysis (Nelson et al. 2004).

Bo and Luo combined support vector machine and VSM (Cosine similarity) to design a personalized recommendation algorithm for using users' profiles (Bo & Luo 2007).

Hsieh *et al*. proposed a personal document recommendation system to reduce the online computation. They used Gini index to identify the most discriminative terms then the terms used represent each document as a vector using TF-IDF. Finally they use their own clustering algorithm which is mainly derived from K-Means. Then the recommendation system recommends a document to the reader which is closer to the document in the same cluster which is read by user currently based on the distance measured among the documents which are projected as vectors (Hsieh et al. 2004).

Lai and Liu combine Knowledge Flow (KF) mining and KF based recommendation to design a document recommendation system. In KF mining phase they represent documents as n-dimensional vector comprised of significant terms, using TF-IDF, then the documents are clustered using Cosine similarity. This phase is used to identify worker's knowledge flow. Then in second phase, they apply a hybrid sequential rule on the target worker. in General The proposed hybrid recommendation methods combine a KF-based sequential rule (KSR) method with a user-based/item-based collaborative filtering (CF) (Lai & Liu 2009).

Hamuda and Kamel present a semi-structured phrase-based document similarity model which indexes web documents based on phrases rather than single term only. This model identifies potential phrases which match between documents and it indicates strong similarity between the documents (Hammouda & Kamel 2002). Chim and Deng also present phrase-based document similarity based on Suffix Tree Model in an efficient way (Chim & Deng 2008). Lebanon uses Riemannian geometry associated with differentiable manifold and set of points to measure the distance between documents based on the provided data. In general the approach is related to maximum likelihood under a model which assigns probabilities inversely proportional to Riemannian volume element (Lebanon 2006). Zhang *et al.* present a similarity measure space model for document clustering. The model derives low dimensional semantic subspace of documents corresponding to the same semantic by maximizing and minimizing the correlation between the documents in the local patches and outside these patches respectively (Zhang et al. 2012).

Chapter 3

Background

There are several algorithms used for measuring similarity between documents such as Pearson (Kornbrot 2005), Spearman (Zar 1998), Kullback-Leibler divergence (Kullback & Leibler 1951), Jaccard coefficient (Jaccard 1912), Shannon (Wartena & Brussee 2008), Euclidean Distance (ED) and Cosine similarity (Salton & Buckley 1988). In this section, first we explain the pre-processing techniques to prepare the documents. Then we describe how the Vector Space Model forms vectors from body of texts. In continue we explain how traditional geometric similarity measures namely Cosine similarity and Euclidean distance use vectors associated to each document to identify the similar and dissimilar documents/texts, finally, we will review some non-geometric similarity measures. The main purpose of this study is to focus on geometric methods. Though the results of many studies (Salton & Buckley 1988, Nelson et al. 2004) show the geometric and VSM based models are more rebuts in measuring similarities among documents compared to non-geometric models, there are not many work focusing on geometric methods to boost these similarity measures for better results. That is why in this study we focus on geometric methods.

3.1 Text Mining

Text mining is a process of discovering unknown information, using extracting information automatically from different written resources. The main issue is to link the extracted information together in order to construct the new facts or hypothesis for further exploration by more conventional means of experimentation. Text mining is different from what search engines do. Principally search helps user to look for something that is already known. In the other words in search it has been tried to push aside all the irrelevant materials in order to find the relevant information .In text mining, the goal is to discover information which are not known from before, something that no one yet knows and so could not have yet written down (Hearst 2003). In fact using text mining, information can be extracted to derive abstract of the words which exist in the documents or to compute that abstracts for the documents based on the words contained in them. Hearst (Hearst 2003) describes the Text mining as a different type of data mining that attempts to find out interesting patterns from huge set of data. The main issue that differentiates regular data mining from text mining is that in text mining the patterns are extracted from natural language text rather than from structured databases of facts. Databases are designed for programs to process automatically; text is written for people to read. Simply saying, there is no program that is able to read texts and it does not seem that will be any program with that ability in future. Many researchers think it will require a full simulation of how the mind works before writing a program that reads the way people do. Text mining also can be called as a process of "numerizing" text. It means, all found words in the input documents will be indexed and counted in order to compute a table of documents and words such as matrix of frequencies that enumerates the number of times that each word occurs in each document. In continue this process can be improved by excluding certain common words such as "the" and "a" (stop word lists). When all the unique words and keywords derived from the document listed other techniques such as standard statistical and data mining can be applied (*Text Mining (Big Data, Unstructured Data)* 2015). In general, text mining involves the application of techniques diverse areas such as information retrieval, natural language processing, information extraction and data mining. As it is shown in Figure 3.1, text mining has three main steps namely Data Selection, Text Preprocessing, Feature Selection and the final product will be bag of words (The Text Mining Process) (Heidarian 2011). In continue the alternative, relevant techniques and methods which fit better to this project are described in detail.

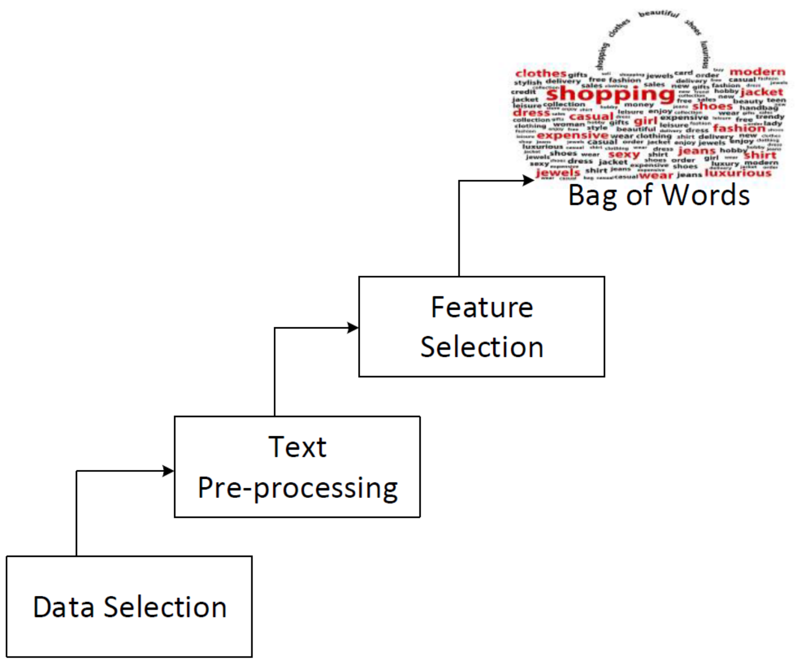


Figure 3.1: Text mining process.

3.1.1 Data Selection

In this step, the relevant data for mining should be identified and prepared. In any database or document, might be many irrelevant data that we do not need to look through them. In order to prevent mining the sparse input, in this step the relevant texts should be identified to avoid mining the redundant data.

3.1.2 Text Pre-processing

Text preprocessing transforms text into an information-rich. During this stage, feature extraction is used in order to extract specific bits of information (*Text mining, also known as intelligent text analysis, text data mining or knowledge-discovery in text (KDT)* 2014). Text preprocessing is done by different methods but the method which is in the scope of this work is Tokenization. Tokenization attempts to explore words in a sentence. Textual data is only a block of characters at the beginning. All processes in information retrieval require the words of the dataset. In fact tokenizer splits up a string of characters into a set of tokens such as words, punctuations, multi-word expressions, phrases, symbols, or other meaningful elements. On the other hand there are still some problems which should be taken into consideration like the removal of punctuation marks, other characters like brackets and hyphens which is done in the next step.

3.1.3 Feature Selection

Feature selection is the study of algorithms for reducing dimensionality (feature reduction) of data to improve machine learning or data mining performance. The objective of feature selection is to remove irrelevant and/or redundant features and retain only relevant features (Sammut & Webb 2011).

Kannan and Ramaraj describe the main purpose in feature-selection methods is to reduce the dimensionality of the dataset by removing some features that are considered irrelevant for the classification. Feature selection step plays a crucial role in text classification. In a set of documents there are thousands of unique words and many of them are not useful for classification. Restricting the set of words that are used for classification makes classification more precise and efficient. Feature selection brings number of advantages such as smaller dataset size, smaller computational requirements for the text classification algorithms and significant shrinking of the search space. Feature selection is a process of selecting a subset of original features according to certain criteria by reducing dimensionality. It reduces the number of features, removes irrelevant, redundant, or noisy data (Karman & Ramaraj 2008). Noisy unstructured text data can be seen in informal settings such as online chat, SMS, email, message board and newsgroup postings, blogs, wikis and web pages.

These types of texts may contain, spelling errors, abbreviations, non-standard terminology, missing punctuation, misleading case information, as well as false starts, repetitions, and pause-filling sounds such as "uhum" and "ah" in the case of speech, stop-words, linking words and emotion characters such as :) ;) :P (Heidarian 2011). Removing unstructured noises is simply done by defining a set of predefined list and remove any string or character which exists in the pre-defined list. Some noisy data are structured such as email addresses or links which follow the particular patterns. In order to remove structured noisy data, patters are defined using Regular Expressions to find the noises and remove them.

One of the important steps in reducing the dimensions of the important terms is stemming. The stemming is used to reduce the inflected words to their word stem, base or root. It is a essential technique for each engines to find the documents/web pages which contain the words with the same stem. For instance if *stemmer*, *stemming*, *stemmed* are given to a stemmer for English language, the output would be *stem* for all of them. Another example, words like: *amusement*, *amusing* and *amused* would be reduced to *amus*. (*Stemming* 2015). Figure 3.2 (Rendón et al. 2011) shows the stemming layers of derived words from *absorb*. Some stemmed words from the datsets used in this study can be found in Table 6.1.

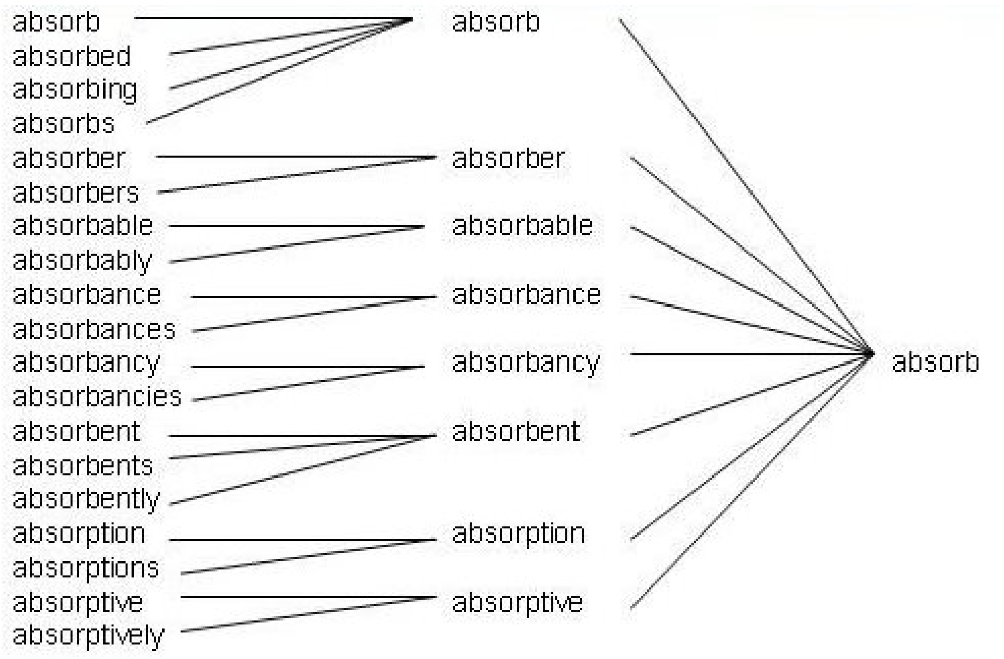


Figure 3.2: Stemming layers of words derived from *"absorb"*.

3.1.4 Bag of Words

After tokenization, removing noisy data and stemming, each document is represented as a set of words, regardless of grammar and order of words in the document. This type of representation is called bag of words which is widely used in document classifications where the frequency of the words plays a crucial role for training and classification (Figure 3.1). In fact words in bag of words represent the most important and significant keywords of each text which play crucial role in measuring the text similarity. In Section 3.2 it will be explained in more details how a document's bag of word containing n keywords is represented as n dimensional vector which form the basis of similarity measures in this study.

3.2 Vector Space Model

Vector Space Model (VSM) is an algebraic model widely used in information retrieval and data mining. The model uses natural language processing techniques to represent each documents/texts as set of vectors with addition and scalar multiplication which introduced by Salton (Salton & Buckley 1988). Each vector describes an object (documents and corpus of texts) using n-dimensional vectors which each dimension representing the frequency of a certain term in a document using a term weighting model called as TF-IDF model:

$$\text{TF-IDF}(d, t) = \text{TF}(d, t) \cdot \log \left(\frac{D}{\text{df}(t)} \right) \tag{3.1}$$

Where TF(d,t) is the frequency of the term t within the document d, and D is the total number of documents and df(t) is the number of documents which contain the term t. The TF-IDF model is a statistical model to show how important a word is to a document/text. The main concept of the TF-IDF relies on a fact that if a word appears rarely in a text, it has a higher importance the text then a word which repeats several times in the text or even in the other texts. For instance some words like "for", "the", "and" which are repeated in each texts for many times are less important then other words which appear rarely. In other words, based on TF-IDF, a weight assigns to term t in document d signifies:

1. The weight is highest when term t repeats many times within a small number of documents (thus lending high discriminating power to those documents).

2. The weight is lower when term t occurs fewer times in a document/text, or occurs in many documents (thus offering a less pronounced relevance signal).

3. The weight is lowest when term t occurs in almost all documents/texts (Manning et al. 2008).

We clarify the model by the following example (Manning et al. 2008). In our example, we have three short texts as following:
d1: *"new york times"*.

d2: *"new york post"*.
d3: *"los angeles times"*.
As we have three documents D=3 and the IDF values are calculated as following:

IDF(angles):	$\log_2(3/1)=1.584$
IDF(los):	$\log_2(3/1)=1.584$
IDF(new):	$\log_2(3/2)=0.584$
IDF(post):	$\log_2(3/1)=1.584$
IDF(times):	$\log_2(3/2)=0.584$
IDF(york):	$\log_2(3/2)=0.584$

Table 3.1 shows the TF scores for all documents.

Table 3.1: The TF score for terms in d1, d2, d3 and d4.

	angeles	los	new	post	times	york
d1	0	0	1	0	1	1
d2	0	0	1	1	0	1
d3	1	1	0	0	1	0

Finally TF values are multiplied by IDF and results are shown in Table 3.2.

Table 3.2: The TF-IDF weights of words in d1, d2, d3 and d4.

	angeles	los	new	post	times	york
d1	0	0	0.584	0	0.584	0.584
d2	0	0	0.584	1.584	0	0.584
d3	1.584	1.584	0	0	0.584	0

In order to illustrate the purpose of TF-IDF and VSM, we have visualized d1, d2 and d3 vectors based on the coordinates gained from table TF-IDF in Figure 3.3.

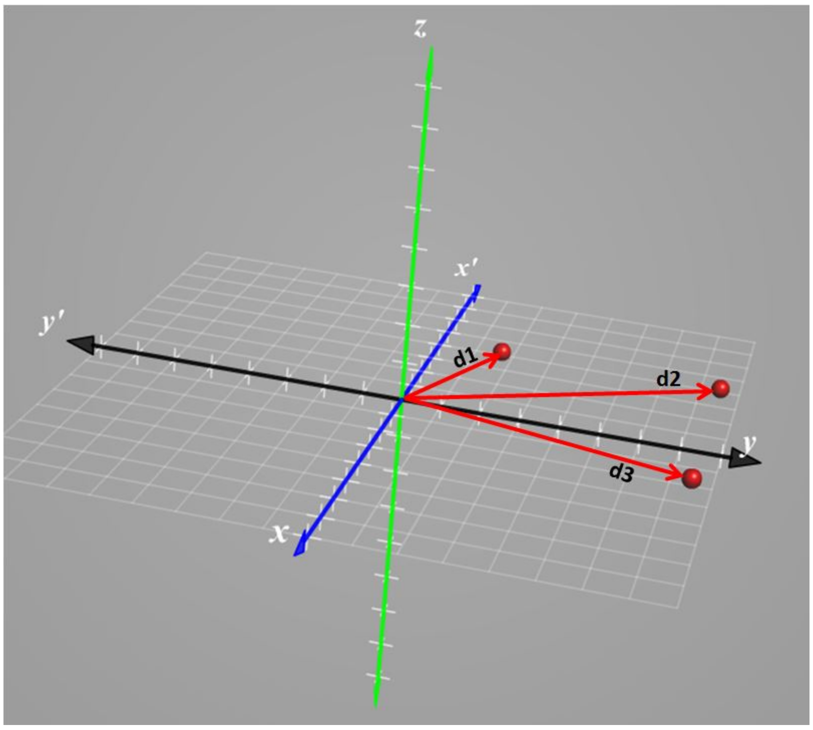


Figure 3.3: Visualizing VSM for d1 ,d2 and d3 in 3D planner.

One of the main drawbacks of TF-IDF is its lack of accuracy due to length of the documents. Long documents have higher term frequencies as they repeat the same term more often (Singhal et al. 1996). The main solution to this problem is different TF normalization components such as Euclidean normalization that dampen the quantity of TF significantly to a unified length (Das et al. 2009, Singhal et al. 1996) and may strongly affect on measurements in a negative way in some cases (Strehl et al. 2000) which will be explained later. In order to use the term frequency in similarity measurements, we do not decrease or dampen the term frequencies and their distributions because they play a crucial role in our hybrid similarity computation. To handle the mentioned drawback, we use TF-IDF Ranking algorithm (Wu et al. 2010) to avoid TF-IDF to bias toward long sentences:

$$\mathrm{TF-IDF}(u,t) = \frac{\mathrm{TF}(d,t)}{W_d} \cdot \log\left(\frac{\mathrm{D}}{\mathrm{df}(t)}\right) \tag{3.2}$$

where W_d is the total number of words in document d. For better understanding o how TF-IDF can be used for similarity measure, let us look at a 2-dimensional example. Let us assume a user who is interested in measuring the similarity among the following three documents with respect to two words namely *"Learning"* and *"Life"*:

- **Document 1:** The game of life is a game of everlasting learning.

- **Document 2:** The unexamined life is not worth living.
- **Document 3:** Never stop learning.

Using the TF-IDF formula, the scores for the three documents are measured in Table 3.3.

Table 3.3: TF-IDF scores for 3 sample documents w.r.t. *Life* and *Learning* keywords.

	Document 1	Document 2	Document 3
Life	0.140550715	0.200786736	0
Learning	0.140550715	0	0.468502384

The measured scores are used as dimensions and the vectors are represented as in Figure 3.4.

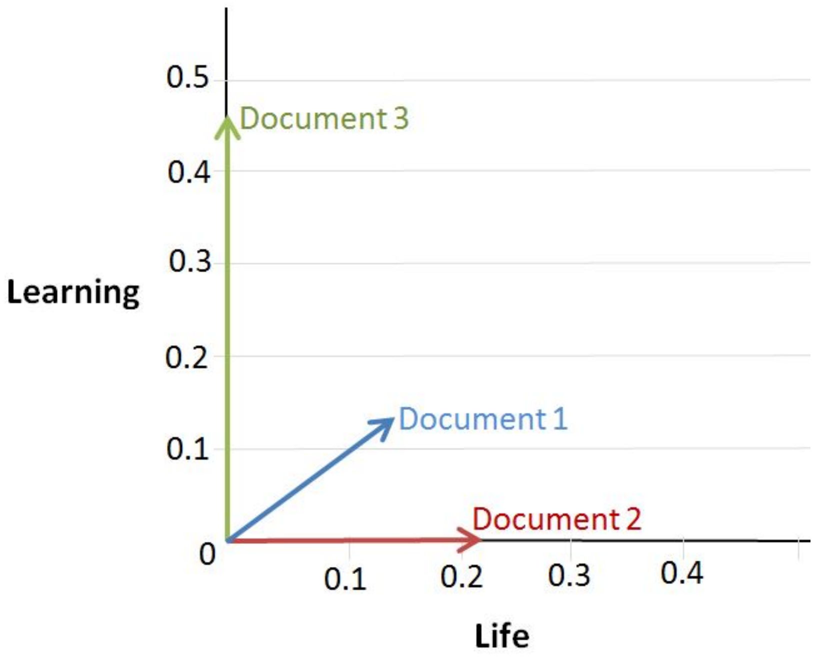


Figure 3.4: VSM for three documents w.r.t. “Life Learning” query.

In Section 3.3 we explain how the VSM can be used to determine the similarity among the three documents projected in Figure 3.4.

3.3 Geometric Measures

After representing documents/texts as vectors using VSM explained in Section 3.2, two traditional similarity measures, namely Cosine similarity and Euclidean distance, are widely used to identify the similar documents. In this section we explain how these two methods measure the similarity among documents.

3.3.1 Cosine Similarity

Cosine similarity computes the pairwise similarity between two documents using dot product and magnitude of vector document A and vector document B in high-dimensional space (Salton & Buckley 1988). This measure helps to avoid the bias caused by documents' length as it uses the angle between two vectors to compute the similarity and has nothing to do with the length of the vectors. In fact, the direction of vectors play the crucial role when the similarities are measured using Cosine method. The inner product of each pair of vectors (sum of the pairwise multiplied elements) is divided by the product of their vector lengths. This approach causes the vectors get normalized to unit length and only the direction of vectors or more precisely the cosine of the angle between each pair of vectors are considered in measuring the similarities. The following formula calculates the Cosine similarity between vector (document) A and vector B in n dimensional space:

$$\text{Cosine}(A, B) = \frac{\sum_{d=1}^{n} A(d) \cdot B(d)}{|A| \cdot |B|} \tag{3.3}$$

The resulting similarity ranges from minimum 0 to maximum 1. That is, if the degree between A and B is 0, it means two vectors are overlapped, in this condition two documents have the maximum similarity and its result is 1 (Cosine 0=1). The measured similarity among three documents mentioned in Figure 3.4 w.r.t. "*Learning Life*" query are in Table 3.4 and shown in Figure 3.5 respectively.

Table 3.4: Cosine similarity among the three documents mentioned in Section 3.2 w.r.t. two keywords in "*Learning Life*" query.

	Document 1	Document 2	Document 3
Document 1	1	0.7	0.7
Document 2	0.7	1	0
Document 3	0.7	0	1

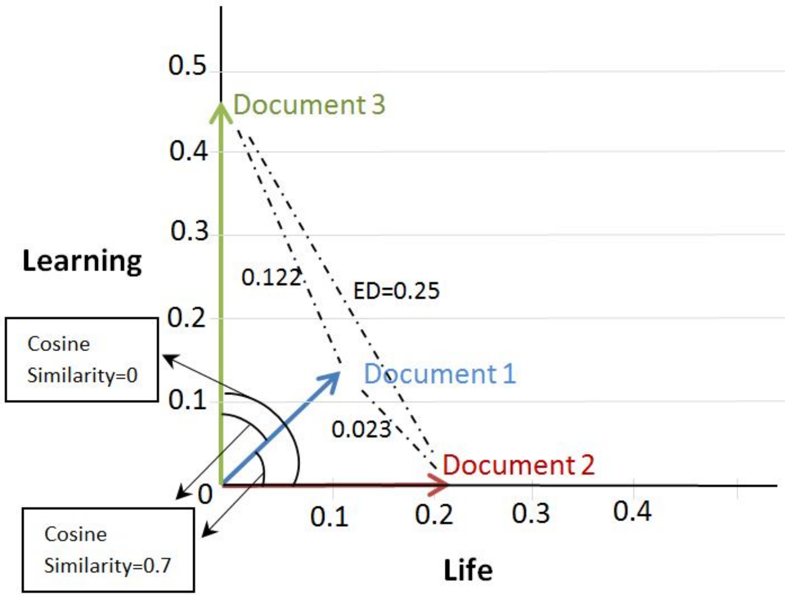


Figure 3.5: Cosine similarity and Euclidean distance measures among the three documents mentioned in Section 3.2 w.r.t. two keywords in "*Learning Life*" query.

3.3.2 Euclidean Distance

Euclidean distance (ED) is another geometrical measure used to measure similarity of two documents. Each document is represented as a point in space based on term frequency of n terms (representing n dimension). ED computes the difference between two points in n-dimensional space based on their coordinate using following equation:

$$\mathrm{ED}(A,B)=\sqrt{\sum_{d=1}^{n}(A(d)-B(d))^2} \tag{3.4}$$

Using ED the highest similarity between two vectors happens when they are plotted in the same point in space and ED between them is 0. Overall, among geometric measures on tasks which are involved in text similarity, Cosine was the best measure (Salton & Buckley 1988, Nelson et al. 2004). The ED among the three documents mentioned in Section 3.2 w.r.t. two keywords in "*Learning Life*" query are in Table 3.5.

Based on the Cosine similarities and ED in in Table 3.4 and Table 3.5 for the three documents, Cosine shows Document 1 has the equal similarity to Document 2 and Document 3, while the results from ED show Document 1 is more similar to Document 2 rather to Document 3. That is the main concepts which can be used for recommendation systems. If a measure is not robust enough to show which documents are more or less similar, the measure could not be reliable for recommendation systems and even it might affect the accuracy of document clustering. In this study we will show when the failure in showing the more and less similar documents changes the clustering accuracy.

Table 3.5: Euclidean distances between each pair of the three documents mentioned in Section 3.2 w.r.t. two keywords in *"Learning Life"* query.

	Document 1	Document 2	Document 3
Document 1	0	0.023	0.122
Document 2	0.023	0	0.25
Document 3	0.122	0.25	0

3.3.3 Other Geometric Measures

There are different types of measuring similarity/difference between vectors and between probabilities such as Manhattan Distance, Hamming Distance, Soergel Distance, Tanimoto Distance and Dice Similarity which are widely used in different fields such as anthropology, biology, chemistry, computer science, ecology, information theory, geology, mathematics, physics, psychology and statistics (Cha 2007). Cha has done a comprehensive survey on distance/similarity measures between vectors and probabilities (Cha 2007). Some measures like Jaccard, Tanimoto and Dice are highly similar and use the same concept to measure the similarity/difference between probabilities. In Section 3.4 Jaccard is explained in brief. Hamming distance is used in Levenstein model in IR which counts the number of changes required to convert a vector/string into another one. For instance if vector A=1,0,0,1,1 and vector B=0,0,1,0,1, then Hamming-Distance(A,B)=2. By changing only two dimensions/elements, vector A can be converted into vector B. Levenstein Distance uses exactly the same concept to measure the similarity between two strings like Hamming-Distance("abc","acb")=2. As trying all the available similarity measures used in different fields is beyond the time limitation of this study, we analysed only ED and Cosine which are widely used in IR. Here we briefly look at Manhattan Distance which is also used in IR.

Manhattan Distance (known as ED)

The Manhattan distance measures the grid-like path travelled from one data point to the other one.This measure is also known as ED, and has the same output on similarity and clustering as ED does. In fact it is a distance between two points measured along axes at right angles (Figure 3.6). It takes the sum of the absolute values of the differences of the coordinates as shown in Equation 3.5:

$$\text{Manhattan}(A, B) = \sum_{i=1}^{k} \big|A(i) - B(i)\big| \tag{3.5}$$

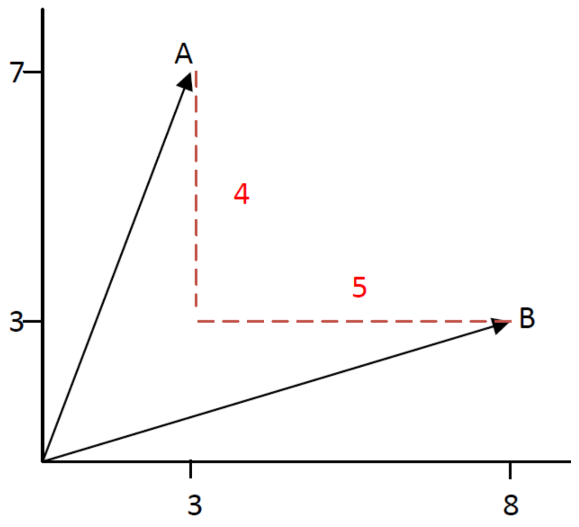


Figure 3.6: Manhattan distance between A and B is $5 + 4$.

3.4 Non-Geometric Measures

There are many non-geometric measures for document clustering and document recommendation, some of them mentioned in related work. In this section we look at the most significant ones briefly, though this study aims to look at geometric measures and evolve the geometric methods by covering their drawbacks. Pearson correlation coefficient is a statistical model to measure the strength of a linear relationship among the desired data. In data mining it is used to compute similarity between two variables (documents or keywords) bounded to -1 and +1. Coefficient 1 shows correlation is positive and two data objects are correlated perfectly, -1 indicates total negative correlation (Kornbrot 2005, Zar 1998).

Jaccard Coeficient (Jaccard 1912) divides intersection of the objects by their unions. The produced coefficient ranges between 0 and 1. If two documents are same, the coefficient is 1 (means they share exactly the same keywords with the same frequency of each) and it is 0 if there is no similarity between them.

Kullback-Leibler Divergence (KLD) (Kullback & Leibler 1951) measures differences between two probability of distributions. It makes the automatic use of term sets for each category. Hence only those terms which belong to predefined category-term lists are taken into consideration and they are compared with each category term probability distribution. It means when a document contains only limited number of terms in comparison to the number of words in categories, the term frequency of many terms in that document is zero (Bigi 2003). Moreover it is asymmetric measure. Jensen-Shannon divergence (also known as Information Radius) is based on the KLD with the difference that it is symmetric (Wartena & Brussee 2008). However Pearson, Jaccard and KLD do not consider document/collection frequency and based on the

TF-IDF concept, rare terms in a collection are more informative than frequent terms (Nayak & Raghavan 2014, Salton & Buckley 1988).

3.5 Clustering

Clustering divides data into groups (clusters) that are meaningful, useful, or both. If meaningful groups are the goal, then the clusters should capture the natural structure of the data. Using clustering, data objects can be grouped based on information which are found in the data that describes the objects and/or the relationships among them. The purpose is to group a similar objects which are related to each other and are different from the other objects in other groups. The more similarity within a group, the more differences between groups and more accurate and distinct clustering (Tan et al. 2013). There are enormous number of techniques and algorithms for document and text clustering, but in general two main approaches are agglomerative hierarchical clustering and K-Means (Steinbach et al. 2000). Steinbach *et al.* have done a comprehensive comparison between these two main techniques and conclude K-Means performs better clustering. They also state even with many runs of K-Means, it is still significantly quicker than a single run of a hierarchical clustering algorithm, particularly if the data sets are large. That is why for this study we decided to choose K-Means for clustering. The method will help us to examine how accurate the similarities are measured and how many percent of the documents in each cluster are really similar using purity method which will be explained in details.

3.5.1 K-Means

K-Means is supervised learning algorithm that is widely used for clustering plotted data points in space (MacQueen et al. 1967). The algorithm clusters data points into K clusters based on K given centroids. In order to get the most discriminated clusters, centroids should be placed as far as possible from each other. Then each point belongs to the nearest centroid to form a cluster. When all nodes are assigned to a centroid, we need to re-calculate K new centroids as new centre of the clusters resulting from the previous step. After K new centroids identified, new assigning should be done to the same data set points to the nearest new centroid (Figure 3.8 and Figure 3.7). The algorithm tries to minimize squared error function mentioned in Equation 3.6 (*K-Means Clustering* 2015):

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left| x_i^j - c_j j \right|^2 \tag{3.6}$$

The K-Means algorithm is accomplished in the following 4 steps:

1. Locate K centroids as initial points of clusters.

2. Assign each data point to the group which has the closest centroid.
3. After assigning all points, recalculate the position of the K centroids.
4. Repeat step 2 and 3 till centroids are stabilized.

The fourth step leads to forming the groups which the metric to be minimized should be calculated using Equation 3.6.

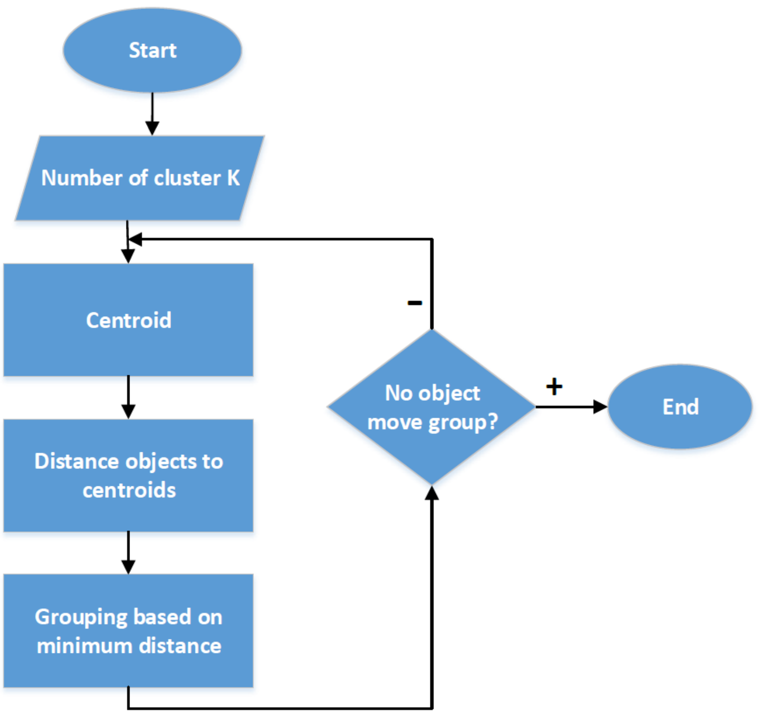

Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
End

Figure 3.7: K-Means flowchart.

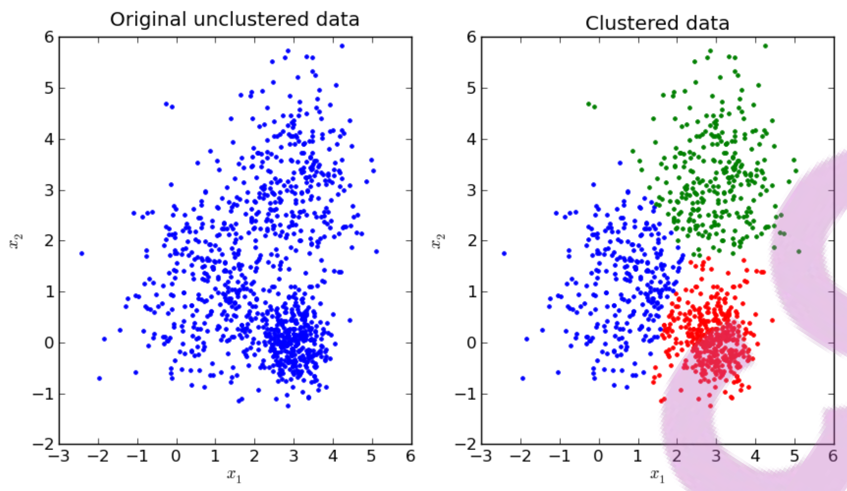


Figure 3.8: Set of sample points after applying K-Means divided into three clusters.

In some studies (Hsieh et al. 2004), clustering using K-Means is used in recommendation systems to find the highly similar objects as described in Section 2). In this study we use the same approach to cluster documents for two purposes, first to prove the accuracy of the proposed method, second to use it as a recommendation solution.

3.5.2 WEKA

One of the popular tools for applying K-Means is WEKA. WEKA is an open source tool written in Java developed at the University of Waikato in New Zealand which contains the comprehensive collection of machine learning algorithms for data mining tasks. The algorithms can be applied on the datasets from GUI or using Java bash scripts from command prompt/bash for heavy tasks. Datasets should be pre-process and converted into ARFF (Attribute-Relation File Format) file which is an ASCII text file that describes a list of instances sharing a set of attributes (*Attribute-Relation File Format (ARFF)* 2008). ARFF files have two sections namely Header and Data. The header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types. The following is an example header on the standard IRIS dataset available in WEKA package:

```
@RELATION IRIS

@ATTRIBUTE sepallength NUMERIC

@ATTRIBUTE sepalwidth NUMERIC

@ATTRIBUTE petallength NUMERIC
```

```
@ATTRIBUTE petalwidth NUMERIC

@ATTRIBUTE class Iris-setosa,Iris-versicolor,Iris-virginica
```

The following is a part of Data section of the same dataset:

```
@DATA

5.1,3.5,1.4,0.2,Iris-setosa

5.1,3.5,1.4,0.2,Iris-setosa

4.9,3.0,1.4,0.2,Iris-setosa

4.7,3.2,1.3,0.2,Iris-setosa

4.6,3.1,1.5,0.2,Iris-setosa
```

K-Means can be used using two types of distance namely Euclidean distance and Manhattan distance (these distance are explained in detail in Section 3.3). For this study we use the most usual one, which is Euclidean distance. Bare in mind, the ED used in K-Means is a separate measure than the one used for measuring similarities. Here we explain some of the parameters which we have modified them for our experiments:

Maximum Iteration (`I`): Maximum number of times the K-Means iterates in the loops explained earlier. We set it to 500 based on the usual number used in other research.

Number of Clusters (`N`): For document clustering, the number of clusters is set as number of the document types in dataset. For instance for 20NewsGroup dataset, we have 20 different types of documents for 20 diverse fields of news. An ideal algorithm clusters the documents in 20 clusters where all documents in each cluster are from the same type, though based on our knowledge there is no such an algorithm capable of clustering any document/textual dataset with 100% accurate clustering.

Seed (`S`): This is a random number of centroids randomly located in far distances from each other.

As the K-Means is a stochastic algorithm, we run the algorithm on each dataset for 100 times on random number of centroids. We will explain about the experiments later. For this study we did not use the GUI of WEKA. We used Java within bash scripts to run K-Means 100 times on each dataset for each measure. The following line is an example of K-Means bash script for WEKA:

```
   java -classpath weka.jar weka.filters.unsupervised.attribute.AddCluster
-W weka.clusterers.SimpleKMeans -N 20 -A weka.core.EuclideanDistance -R
first-last -I 500 -S 25 -i inputFile.csv -o outputFile.csv
```

3.6 Clustering Evaluation

After applying any clustering method, evaluating the quality of the clusters is the first way to examine the reliability of the method. Clustering validation can be done via Internal validation and External validation. Internal validations is used when there is no priori information available about the data and it can be done based on the clustering structure. Some internal validity measures and indices are Cohesion and Separation, Silhouette Coefficient, Bic index, Calinski-Harabasz index, Davies-Bouldin index (DB), Dunn index and Niva index (Rendón et al. 2011).

External validation indexes is suitable for situations where we have a priori knowledge of dataset information. In this study we apply the external clustering method, because we do the experiments on the labelled datasets which can be used to evaluate the quality of clusters. Some external validity indices are F-measure, Purity and Entropy. In this section we explain how they work, though we have used Purity in this study in order to evaluate the quality of the clusters.

3.6.1 F-Measure

This method combine precision and recall concepts. The precision and recall for each class is calculated as following:

$$\begin{aligned} \text{Recall}(i,j) &= \frac{n_{i,j}}{n_i} \\ \text{Precision}(i,j) &= \frac{n_{i,j}}{n_j} \end{aligned} \tag{3.7}$$

where $n_{i,j}$ is the number of documents of class j that are in cluster j, and n_j is the number of documents in cluster j, and n_i is the number of documents in class i. Then the F-Measure of cluster j and class i is calculated as the following equation:

$$\text{F}(i,j) = \frac{2 \cdot \text{Recall}(i,j) \cdot \text{Precision}(i,j)}{\text{Precision}(i,j) + \text{Recall}(i,j)}$$

The results of F-Measure varies between 0 and 1, when 1 indicates the highest clustering purity (Rendón et al. 2011).

3.6.2 Purity

Purity is a simple evaluation measure which is very similar to Entropy. To compute the purity, first each cluster is assigned to the class which is the most frequent in the cluster. Then in order to compute the accuracy of the assignment, we divide the number of correctly assigned documents by total number of documents in the cluster, in ideal situation, if there are N documents in cluster C and all of them are from the same class, the purity is 1 which is the highest cluster quality. The purity is measured using the following formula:

$$P(C_i) = \frac{d_i}{n_i} \tag{3.8}$$

where n_i is the size of the cluster C_i and d_i is the number of documents from dominant class in cluster C_i. As mentioned earlier, K-Means is a stochastic algorithm and because of that we need to run K-Means 100 times. For each run we measure the purity of each cluster and, in order to show how close the purity results of each cluster are, the highest purity value is selected and standard deviation of purities of all clusters is calculated to show how scattered the purity results are. So if by chance a measure generates few high quality clusters and several low quality clusters, sigma operation would give the high purity result which do not reveal the reality behind the all purities, while standard deviation shows how steady a measure clusters documents with the approximately same purity in majority of clusters. Let us see how purity shows the quality of a cluster in a toy example. For instance, in 20NewsGroup we have 20 types of news, here we pick randomly some documents from three types of news namely Politics, Economics and Sports and throw them into cluster C as show in Figure 3.9. In this example, there are seven documents from Sports category, so Sports is the dominant class. Now the purpose is to evaluate how many percentage of cluster C contains documents from the dominant class. As the total number of documents in cluster C is 14 and 7 of them are from the dominant class, so the purity of the cluster is 0.5. The purity results varies between 0 and 1 where 1 shows the highest quality cluster.

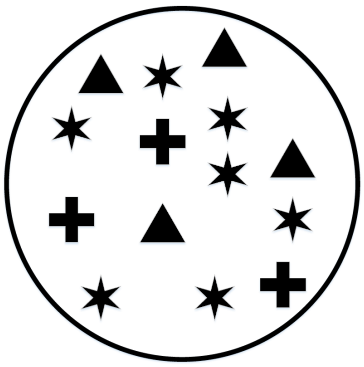


Figure 3.9: $\triangle$=Economics, $+$=Politics, $\star$=Sports.

3.6.3 Entropy

The main concept of the Entropy is similar to Purity. It measures the purity of the clusters based on the class distribution of the documents in each cluster as following:

$$\mathrm{E}_j = \sum_j p_{i,j} log(p_{i,j}) \tag{3.9}$$

where the sum is taken over all classes. The total entropy for a set of clusters is calculated as following:

$$\mathrm{E} = \sum_{j=1}^{m} \frac{n_j}{n} \mathrm{E}_j \tag{3.10}$$

where n_j is the size of cluster j, m is the number of clusters, and n is the total number of data points.

Chapter 4

Motivation

The main purpose of this study is to measure the similarities among documents with high accuracy in such a way that one hopes to better understand which documents are more similar (or less similar). We call this concept similarity level focused in this study. In this study, *accuracy* refers to the power of a measure in differentiating the similarity level among documents in such a way that one can understand which documents are less, more and most similar (refer to Figure 4.1 and Figure 4.2 and the related details explained). This power of differentiation can be significantly useful for recommendation systems and clusterings. As explained earlier, similarity level is a helpful measure in recommendation systems and document clustering. As explained more in details earlier, this study focuses on geometric similarity measures which are popular for document clustering but there are not much work on improvement of the current geometric models in such a way that can be used for measuring a concept called *similarity level.* In some research and surveys (Nelson et al. 2004, Salton & Buckley 1988), diverse similarity measures used for IR have been evaluated and results show Cosine similarity outperforms other measures, that is another reason to focus on geometric measures like ED and Cosine to enhance them in such a way that can be used to measure the similarity levels. The applicability of the similarity level will be explained more in details in next sections. In this section we explain why existing geometric measures are not robust enough to measure the similarity level accurately.

4.1 Cosine Drawbacks

Cosine similarity is a powerful method for measuring the difference between two documents based on their orientations but not their magnitudes. Hence magnitude of vectors which is dealing with term frequency does not play any role in this similarity measurement.

4.1.1 Uniqueness issue

Figure 4.1 shows two critical situations wherein Cosine similarity's weaknesses may produce inaccurate similarity results toward the vectors' magnitudes. Based on Cosine similarity metric, the similarity between document A and document B, document A and document C and finally between document A and document D are equal. Despite the identical angles between them, there is a huge difference between the vectors' magnitudes. As A and B have the higher proportion of the terms in their texts, it seems A and B are more more about the terms from searched query than C and D. Hence A has higher similarity to B, less similarity to C and least similarity to D. That is why Cosine produces many similarity values with the same value rather than producing unique similarity values when there are difference among similarities (uniqueness issue).

4.1.2 Boolean values

The other limitation associated to VSM is Boolean values (Wong et al. 1985). The Boolean values which are more witnessed in Cosine Similarity, gives a generic view over similarity among documents. For instance, in Figure 4.1, based on Cosine similarity, document B has the same similarity of 1 to document C and document D, while it is obvious that B and D are less similar because there is a longer distance between B and D so the difference is higher.

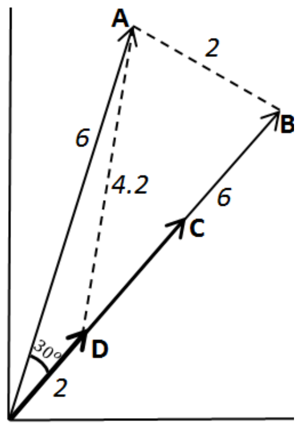


Figure 4.1: Example of drawbacks using Cosine.

Hence Cosine similarity cannot be that proper and accurate method for measuring similarity level between vectors.

4.2 Euclidean Distance Drawbacks

Figure 4.2 shows the main ED's drawback clearly. As it can be seen many vectors such as P, Q and R can be drawn from M with the same ED and despite the huge difference between them, ED shows P, Q and R have got the same similarity of 3 to M. Vector P, Q and R have almost similar magnitudes, even they have the same ED to M as their end nodes are plotted on the edge of the circle drawn from a centre close to 0 coordinate. But ED results show M has the equal similarity to P, Q and R. It is obvious P is more similar to M as it is closer to M. ED does not convey this fact and that is why it can not be used as a reliable method to measure the similarity level.

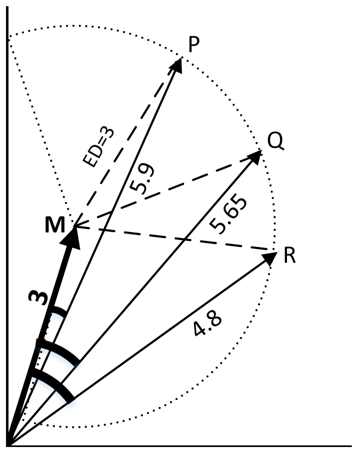


Figure 4.2: Example of drawbacks using ED and Manhattan distance.

4.3 Manhattan Distance Drawbacks

Although Manhattan distance is not popular and seldom used in literature, we briefly explain why it is not a proper measure for measuring similarity level. As show in Figure 4.3, many vectors with the same magnitude like vector T and vector S can be drawn with different distances to a vector like R. If the purpose is to measure the similarity levels in order to identify whether S is more similar to R or T is more similar to R, the obvious answer would be S. Because from magnitude point of view S and T are equal, while from angular point of view S is closer to R. Based on experiments from the literature, Cosine is the most robust measure as it takes the angular differences into similarity measure (Salton & Buckley 1988) while, Manhattan distance does not do so. Likewise ED, Manhattan distance is not able to convey this concept. In this toy example, Manhattan(R, S)=Manhattan(R, T)=5, which means S and T are in the same level fo similarity to R. In some work (Mihalcea et al. 2006) Manhattan distance is considers identical to ED and it gives the same result as ED in terms of similarity measure. That is why we do no focus on Manhattan distance in this study. Even after applying Manhattan distance on vectors in Figure 4.2, to understand which vector has the highest and lowest similarity to vector M, we get the following result: Manhattan(M, P)>Manhattan(M, Q)=Manhattan(M, R). It means Q and R are more similar to M and Q and R have the same similarity level to M, and as we discussed earlier this conclusion conveys the wrong result for similarity level concept.

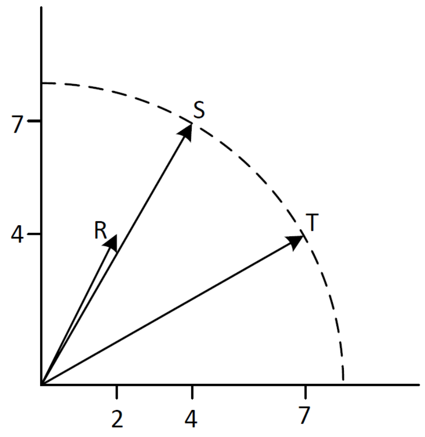


Figure 4.3: Example of drawbacks using Manhattan distance.

Chapter 5

TS-SS: Novel Geometric Similarity Measure Method

As it was mentioned earlier, the similarity measurement has to take underlying semantic features into consideration in order to reflect the meaningful results (Strehl et al. 2000). In the other word, by using the magnitude of vectors as an influential and powerful tool, we can compute similarity among documents more accurate. In order to interpolate the magnitudes into similarity measurement, it is required to include more parameters than the angle and ED between vectors. In this research, a new algorithm called TS-SS computes the similarity between vectors from two divers prospective and generates the similarity value between two vectors not only from the angle and ED between them, but also the difference between their magnitudes. The model is examined on different datasets to prove its accuracy and robustness in clustering and measuring similarity level.

5.1 Triangle's Area Similarity (TS)

By looking at vectors in Figure 5.1, it is obvious that a triangle can be formed as the ED is drawn between them. As the ED between two vectors decreases and they get closer to each other (so the angle between them gets tighter also), the area of the triangle decreases. This decrement has the inverse relation to similarity between two documents.

5.1.1 Calculation and Strength of TS

By calculating the area of the triangle between two vectors and using it as similarity metric, in fact three characteristics have been included in computing the similarity, namely angle between vectors, magnitude of vectors and ED. We call this method TS (Triangle's Area Similarity) and present it as TS(A,B) for two vectors of A and B.

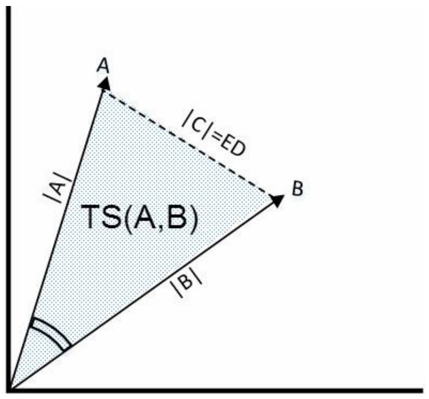


Figure 5.1: Triangle similarity (TS).

We use SAS (Side Angle Side) formula to calculate the area of the triangle. To that end, it is required to measure the magnitude of the vectors and the angle between them: $|A| = \sqrt{\sum_{n=1}^{k} A_n^2}$, $|B| = \sqrt{\sum_{n=1}^{k} B_n^2}$ where n is dimension of vector $(x, y, z, \ldots)$. In order to compute the angle between vectors, first Equation 3.3 is used to get V (the cosine value of θ), then using the following cosine rule, the new angle (θ') is calculated:

$$\theta' = \cos^{-1}(V) + 10 \tag{5.1}$$

The idea of forming the triangle fails when vectors are overlapped like B and C in Figure 4.1. To overcome this problem, we increased θ by 10 degrees. We used 10 degrees as minimum to keep it round and the reason to do not use any value less than 10 is to avoid more decimal points and keep our calculations simple.

$$\text{TS}(A, B) = \frac{|A| \cdot |B| \cdot \sin(\theta')}{2} \tag{5.2}$$

Using this approach, as two vectors get closer to each other and the area of triangle between them decreases, proportionally the dissimilarity decreases and the similarity increases and the maximum similarity is met when two vectors with the same length are overlapped. Unlike ED, using TS method, the measured similarity of M and P, is different with similarity of M and Q in Figure 4.2. That is, TS(M,P) < TS(M,Q) which is an expected result as P is more similar to Q. As the area of triangles among pairs of vectors vary due to differences in their magnitudes and angles between them, TS generates different similarity values.

5.1.2 Weakness of TS

The TS accuracy fails in one of the cases drawn in Figure 4.1. Vectors A and B look more similar as they have got the same magnitude, while D is smaller than A and ED(A,D) is approximately three times more than ED(A,B). Despite these obvious distance differences, it can be seen TS(A,D) < TS(A,B) and it conveys that A is more similar to D rather than to B which may not be an accurate assumption.

5.2 Sector's Area Similarity (SS)

Merely TS alone is not robust enough to interpolate vectors' differentiations precisely to produce accurate similarity results due to missing components. By looking at Figure 4.1, it can be understood that one of the missing components is the difference between the vectors' magnitudes. In this study the magnitude difference between two vectors is called MD and represented as:

$$\mathrm{MD}(A,B) = \left| \sqrt{\sum_{n=1}^{k} A_n^2} - \sqrt{\sum_{n=1}^{k} B_n^2} \right| \tag{5.3}$$

MD has a one way direct relation to ED. That is, as MD increases, proportionally ED increases. Combination of ED and MD might be helpful but in order to leverage their effects on similarity values we need the angle between vectors which is called angular difference and represented as AD(A,B) $= \theta'$ for two vectors A and B. AD can be calculated using Equation 5.1. Like the relation between MD and ED, AD and ED also has the one way direct relation. In the other word, in all cases, increasing and decreasing the angle between vectors effects on ED directly. That is why AD is another important component in measuring similarity. ED and MD can be combined by summation and AD can accentuate the power of this combination by forming a circular section with a radius of ED+MD length (Figure 5.2).

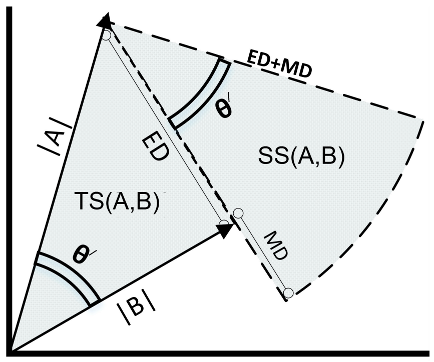


Figure 5.2: TS-SS model.

By measuring the area of the formed section, we can calculate the similarity between two vectors from another prospective. This similarity is called SS (Section's Area Similarity) and computed using the following formula:

$$\mathrm{SS}(A,B) = \pi \cdot \big(\mathrm{ED}(A,B) + \mathrm{MD}(A,B)\big)^2 \cdot \left(\frac{\theta'}{360}\right) \tag{5.4}$$

5.3 TS-SS Method

TS and SS complete each other and that is the reason we combine them by multiplying them together. The range of TS-SS measure is from 0 to ∞. The reason for choosing multiplication but not summation to combine TS and SS is that in some cases the value of TS and SS are disproportionate where one is extremely larger then the other one due to their quadratic calculations with respect to the length of the vector. For example, in Figure 4.1, TS similarity is too big (TS(A,B)=5.91) while SS similarity is too small (SS(A,B)=0.047). If we use summation, they can not effect on each other significantly to give the realistic similarity value and we get TS(A,B)+SS(A,B)=5.95 $>$ TS(A,D)+SS(A,D)=2.96 which is a false result because A is more similar to B as discussed earlier. But if we use multiplication we get TS(A,B)$\cdot$ SS(A,B)=0.27 $<$ TS(A,D)$\cdot$SS(A,D)=1.71 which is a true result. Using TS-SS method, similarity of 0 happens only when ED=MD=0 and it shows two vectors are absolutely identical in term of direction and magnitude which indicates the maximum similarity between two documents. This novel similarity method is called TS-SS and is presented as following:

$$\text{TS-SS}(A,B) = \frac{|A| \cdot |B| \cdot \sin(\theta') \cdot \theta' \cdot \pi \cdot \big(\text{ED}(A,B) + \text{MD}(A,B)\big)^2}{720} \tag{5.5}$$

Chapter 6

Experiments

In this section first we briefly describe the five datasets used in this study. Then we will explain the evaluation models and finally we compare the three metrics based on the results of evaluations. We assume that the end user enters three keywords in search engine and is interested in finding the clusters of similar documents where each document contains at least one of searched query's keyword in a selected dataset. Based on this assumption, at first we apply the data integration and text preprocessing mentioned in Section 3.1 on all texts in each dataset and create a list of top 600 keywords which have the highest TF-IDF score (as mentioned earlier, the higher informative a keyword is, the higher TF-IDF score it has). Some of the selected keywords from the databases are listed in Table 6.1. Words are stemmed to their roots.

Then we create a number of different search queries, each consisting of three keywords picked from the list of top keywords randomly and passed to our search engine. Finally we create pairwise Cosine similarity matrix, ED similarity matrix and TS-SS similarity matrix for each search query in each dataset. After that, we apply K-Means clustering, on the results gained from each measure. As mentioned in Section 3.5.1, K-Means clusters all close data points which are close to a centroid. On the other hand, the documents of each class in each dataset are all next to each other, while in real world, documents have no label and all diverse type of documents are mingled. For instance *Document-1* till *Document-1000* belong to sports news and then the rest of the documents of other news types also come in the same order one after the other. Therefore we need to shuffle the similarity measures gained from all documents, in order to randomly relocate data points. Otherwise K-Means may return an high quality but unrealistic clustering results. A toy example of how each document is assigned to a cluster using WEKA (Section 3.5.2) can be found in Table 6.2. Then we use the labels (type of documents) to verify how many percent of documents in each cluster are from the same class (same type of documents e.g. Sports or Economics).

Table 6.1: Sample top stemmed keywords with highest TF-IDF scores from the four datasets.

20NewsGroup	**Classic4**	**7Sector**	**WebKB**
nntp	airfoil	helicopt	meta
govern	tumor	cameron	eric
univers	clinic	silver	roger
christian	symmetr	rohm	carolina
comput	arbitrari	parkway	london
control	configur	pittsburgh	kenneth
data	plasma	micro	illinoi
object	mix	alabama	xerox
kill	environ	transform	church
armenian	friction	pennsylvania	lawrenc
version	stagnat	tube	mpeg
jesus	nozzl	api	multiscalar
uk	optim	thailand	webcrawl
american	dynam	vacat	encrypt
machin	hormon	nationsbank	maryland
engin	slender	alumax	christian
graphic	cylindr	aerospac	sound
internet	gradient	quist	ijcai
jew	dna	fluid	wagner
religion	wind	accid	lazowska
mac	kidney	carpet	taylor
israel	buckl	venezuela	jackson
ftp	turbul	teradyn	germani
polit	reynold	alzheim	crime
bibl	fluid	ireland	french
fbi	aerodynam	powerwav	uiuc
earth	treatment	amplifi	singapor
monitor	blunt	whatsoev	silicon
drug	error	connecticut	empir
muslim	blood	louisiana	fish
turkish	chemic	timberland	utexa
homosexu	temperatur	temperatur	gvu
devic	speed	medicin	melski
arab	superson	switzerland	brain
jewish	plate	plaza	cpsc
weapon	organ	norfolk	eicken
video	laminar	cajun	king
isra	cylind	harvest	nasa
jim	jet	kidney	quantum
ibm	linear	yahoo	bestavro

Table 6.2: Clustering toy example after applying K-Means on four documents using WEKA.

	Doc-1	Doc-2	Doc-3	Doc-4	Clusters
Doc-1	0	0.2	0.8	0.6	Cluster 1
Doc-2	0.2	0	0.7	0.9	Cluster 1
Doc-3	0.8	0.7	0	0.3	Cluster 2
Doc-4	0.6	0.9	0.3	0	Cluster 2

In order to evaluate and show the correctness of TS-SS measure, we compare similarity matrices from the three geometrical metrics by running four different evaluations. Finally we represent a test case including four documents derived from Classic4 and show the significance of some drawbacks mentioned in Chapter 4 in real world.

6.1 Datasets

The five chosen datasets contain sets of documents. Four of them are labelled by their topics manually and are widely used for data classifications, text categorization and clustering and one of the datasets is a small unlabelled Twitter dataset.

6.1.1 20 News Group

This dataset contains 20 different categories and each category has 1000 newsgroup documents (*The 20 Newsgroups data set* 2008). For this dataset we have computed the similarity values for 20 search queries.

6.1.2 7 sectors

This dataset consists of classified documents from seven industrial sections, and each section has around seven subsections (*CMU World Wide Knowledge Base Project* 2011). Totally there are 44500 labelled documents in this dataset. For this dataset, we have computed the similarity values for ten search queries.

6.1.3 WebKB

This dataset is the collection of web pages collected from computer science department of diverse universities in 1997 and manually classified into seven classes (*WebKB* 2010). This database contains 8334 documents. For this dataset we have computed the similarity values for 10 search queries.

6.1.4 Classic4

The Classic4 dataset contains 7095 labelled documents from abstract of scientific papers in four divers categories (*Classic3 and Classic4 DataSets* 2010). For this dataset we have computed the similarity values for ten search queries.

6.1.5 Twitter

For this study we have selected 30,000 tweets from a Twitter dataset which includes 476 million unlabelled tweets (Yang & Leskovec 2011). As the dataset is unlabelled we can not apply two of evaluations (Purity and minimum Gapscore) as discussed in Section 6.2.

6.2 Evaluations

After computing similarity values for all search queries in each dataset using Cosine, ED and TS-SS metrics, we compare these three metrics by using four different evaluations namely Uniqueness, number of booleans, minimum Gapscore and Purity.

6.2.1 Uniqueness

The purpose of this evaluation is to compute the percentage of unique values in each similarity matrix. The outcome of this evaluation indicates the existence possibility of drawbacks mentioned in Chapter 4.1. When we have more unique similarity values, it means the measure is robust to recognize the similarity level among documents even when there is a small difference among documents, rather than generating same similarity values among different documents.

6.2.2 Number of Booleans

This evaluation counts the number of Boolean values in the similarity matrices generated by each measure and shows how many percent of values are booleans. The main purpose of this evaluation is to show which measure produces more boolean values and gives less variation. In fact this is another attempt to evaluate the highlighted drawbacks in Chapter 4.1.

6.2.3 Purity

Purity is widely used for measuring the quality of clusters based on the label of documents in each cluster. In this study labels identify which category each document belongs.

As described in details in Section 3.6.2, to measure the purity of clusters, each cluster is assigned to the category which is the most frequent in the cluster. Then the assignment accuracy of cluster C_i which contains n_i documents is measured by the following formula (*Evaluation of clustering @ONLINE* 2009):

$$P(C_i) = \frac{d_i}{n_i} \tag{6.1}$$

where d_i is the number of documents from the dominant category in cluster C_i. In order to use the purity metric to identify which metric generates more realistic similarity values, first we need to cluster the values. For that purpose we cluster values based on K-Means algorithm (explained in Section 3.5.1) using WEKA (explained in Section 3.5.2). K-Means is a non-deterministic algorithm and for different number of seeds generates different clusters. In order to get the robust result, we run the k-means algorithm with diverse number of seeds for 100 times on all search queries' similarity matrices and compute the average as the final result.

6.2.4 Minimum Gapscore

In this test we compare the similarity matrices with their associated oracle. As the documents are labelled, we are able to construct the oracle matrix using the existing document labels from the datasets.

For a fixed set of documents, let S be a Boolean Oracle that returns $S[i,j] = 1$ ($=$ true) if and only if document i is similar to document j. For a similarity measure α we define its *gap score* (with respect to S) as

$$\min_{B_c} \sum_{1 \le i < j \le n} \left| \, B_c(\alpha(i,j)) - S[i,j] \, \right|,$$

$$\text{where } B_c(x) = \begin{cases} 1 & \text{if } x \ge c \\ 0 & \text{if } x < c \end{cases}$$

Furthermore, for two α_1 and α_2 we define a quasi order $\alpha_1 \le_S \alpha_2$ if the gap score of α_1 is at most the gap score of α_2. Here we present a method to compute the gap score in Figure 6.1.

Theorem 1. *The Algorithm 6.1 correctly computes the gap score of* similarity *(i.e. α) with respect to* oracle *(i.e. S).*

Proof: The loop at label L1 does two things. First, it tallies up the number of entries of the oracle that are true, which will be the gap score if the cut-off B_c for α does not include any true (=1) cases; this is stored in 'curGapScore'. Second, it makes a vector of pairs of keys—similarity values $\alpha(j,k)$ and the corresponding Boolean 0/1 entry of the oracle $S[j,k]$.

```
Begin
    Input: real similarity[$n$][$n$]; bool oracle[$n$][$n$]
    Sim = vector [];
    curGapScore = 0
L1: For $j = 1$ to $n - 1$ do
        For $k = j + 1$ to $n$ do
            Sim.append(pair(similarity[$j$][$k$],oracle[$j$][$k$]))
            If oracle[$j$][$k$]==1 then
                curGapScore = curGapScore+1
L2: Sort Sim by decreasing order key1 then increasing order key2
    minGapScore = curGapScore
L3: For each (s,b) in Sim do
        If $b$==1 then
            curGapScore = curGapScore−1
        else
            curGapScore = curGapScore+1
        If curGapScore < minGapScore then
            minGapScore = curGapScore
    Return minGapScore
End
```

Figure 6.1: Algorithm for finding minimum Boolean gapscore.

The sort at label L2 orders the similarity values of α in decreasing order and if ties then sorts all false (=0) oracle values before true values.

The iterations through the sorted 'Sim' vector at label L3 will successively lower the cut-off B_c, i.e. the value of c and update 'curGapScore' to be:

$$\sum_{1\leq i<j\leq n} \left| B_c(\alpha(i,j)) - S[i,j] \right|$$

Ignoring ties for now, we claim that after the i-th iteration of loop L3 the largest i similarities of α have been set to true and the variable 'curGapScore' reflects that score. Base case ($i = 0$): on entry to the loop we have no similarity values set to true, so the gap score is correctly set to the total number of true values of the oracle. Inductive case: processing the i largest similarity value, we either decrease the current gap score by one (if that corresponding entry in the oracle is b==1) or we increase the current gap score by one (if that corresponding entry in the oracle is b==0).

We use variable 'minGapScore' to keep track of the best gap score over all possible cut-offs B_c. However, we need to ensure that if there are ties in similarity values that we only update this variable when the (i+1)-th largest similarity is different than the i-th largest. If the second key of 'Sim' is ordered with all falses before trues then the variable 'curGapScore' will initially increase before decreasing; so 'minGapScore' will be updated (if at all) at the final tied similarity value entry of 'Sim'.

Corollary 2. *There exists an algorithm that decides the Gapscore Problem in time $O(n^2 \max(k, \lg n))$, assuming one can compute α_1 and α_2 in linear time (function of k) and the evaluation (Oracle call) of S takes constant time.*

Proof: We can call Algorithm 6.1 two times for $\alpha \in \{\alpha_1,\alpha_2\}$ to determine the best gap scores, respectively for each. The total time complexity of this algorithm may be broken down to three phases:

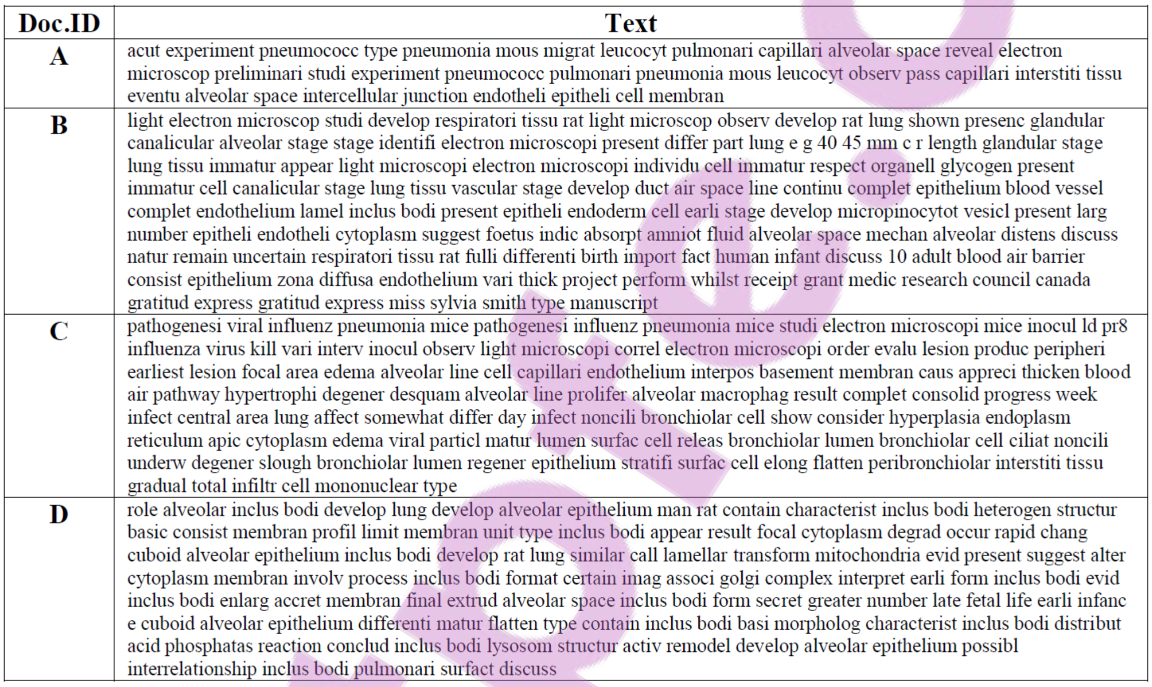
- *Phase one:* Each similarity value for α can be computed in linear time of k; therefore the time complexity for computing all evaluations (e.g. similarity$[j][k]$ for $1 \leq j < k \leq n$) of α is $O(n^2k)$.
- *Phase two:* Building and sorting the vector 'Sim' can be computed in $O(n^2 \lg n)$, as denoted by loop L1 and line L2 of Algorithm 6.1.
- *Phase three:* Find the minimum cut-off value c for B_c over all possible cut-offs takes an additional $O(n^2)$ steps, as done in loop L3 of Algorithm 6.1.

From Phases one to three we conclude total running time of :
$O(n^2k) + O(n^2 \lg n) + O(n^2) = O(n^2 \max(k, \lg n))$.

6.2.5 Test Case

We selected four documents from Classic4 which every one has at least one of the keywords in the search query of *"alveolar aneurysm car"*. For simplicity we have changed the name of the documents to A, B, C and D and the text preprocessing (e.g. stemming, removing stop words and noisy data) has applied on text belonging to each document in Table 6.3. Based on the similarity values, keyword frequencies and number of words in each document, we show the robustness of each measure in computing similarity levels.

Table 6.3: Test case from Classic4.



Doc.ID	Text
A	acut experiment pneumococc type pneumonia mous migrat leucocyt pulmonari capillari alveolar space reveal electron microscop preliminari studi experiment pneumococc pulmonari pneumonia mous leucocyt observ pass capillari interstiti tissu eventu alveolar space intercellular junction endotheli epitheli cell membran
B	light electron microscop studi develop respiratori tissu rat light microscop observ develop rat lung shown presenc glandular canalicular alveolar stage stage identifi electron microscopi present differ part lung e g 40 45 mm c r length glandular stage lung tissu immatur appear light microscopi electron microscopi individu cell immatur respect organell glycogen present immatur cell canalicular stage lung tissu vascular stage develop duct air space line continu complet epithelium blood vessel complet endothelium lamel inclus bodi present epitheli endoderm cell earli stage develop micropinocytot vesicl present larg number epitheli endotheli cytoplasm suggest foetus indic absorpt amniot fluid alveolar space mechan alveolar distens discuss natur remain uncertain respiratori tissu rat fulli differenti birth import fact human infant discuss 10 adult blood air barrier consist epithelium zona diffusa endothelium vari thick project perform whilst receipt grant medic research council canada gratitud express gratitud express miss sylvia smith type manuscript
C	pathogenesi viral influenz pneumonia mice pathogenesi influenz pneumonia mice studi electron microscopi mice inocul ld pr8 influenza virus kill vari interv inocul observ light microscopi correl electron microscopi order evalu lesion produc peripheri earliest lesion focal area edema alveolar line cell capillari endothelium interpos basement membran caus appreci thicken blood air pathway hypertrophi degener desquam alveolar line prolifer alveolar macrophag result complet consolid progress week infect central area lung affect somewhat differ day infect noncili bronchiolar cell show consider hyperplasia endoplasm reticulum apic cytoplasm edema viral particl matur lumen surfac cell releas bronchiolar lumen bronchiolar cell ciliat noncili underw degener slough bronchiolar lumen regener epithelium stratifi surfac cell elong flatten peribronchiolar interstiti tissu gradual total infiltr cell mononuclear type
D	role alveolar inclus bodi develop lung develop alveolar epithelium man rat contain characterist inclus bodi heterogen structur basic consist membran profil limit membran unit type inclus bodi appear result focal cytoplasm degrad occur rapid chang cuboid alveolar epithelium inclus bodi develop rat lung similar call lamellar transform mitochondria evid present suggest alter cytoplasm membran involv process inclus bodi format certain imag associ golgi complex interpret earli form inclus bodi evid inclus bodi enlarg accret membran final extrud alveolar space inclus bodi form secret greater number late fetal life earli infanc e cuboid alveolar epithelium differenti matur flatten type contain inclus bodi basi morpholog characterist inclus bodi distribut acid phosphatas reaction conclud inclus bodi lysosom structur activ remodel develop alveolar epithelium possibl interrelationship inclus bodi pulmonari surfact discuss

Chapter 7

Results And Discussions

As explained earlier, different number of randomly-generated search queries have been used to compute the similarities in each dataset. For each search query, we have three similarity matrices namely Cosine, ED and TS-SS similarity matrix. We applied the evaluation techniques on all matrices and the results shown in following tables represent the average results, except Table 7.3 which contains the best results (maximum purities), though the average, minimum and maximum values also have been plotted in Figure 7.3, Figure 7.2 and Figure 7.1. Also the results presented in percentage in Table 7.1 and Table 7.2 indicate in average, the percentage of similarity values that are unique and Boolean respectively.

7.1 Uniqueness Results

As Table 7.1 shows Cosine similarity has the very low percentage of uniqueness and it conveys that the drawbacks mentioned about this model in Section 4.1 (Figure 4.1) is a significant issue. This low accuracy in terms of recognizing the differences among similarity values causes the lack of distinguishment between documents with higher similarities and documents with lower similarities. Although intangible difference between ED uniqueness and TS-SS uniqueness indicates that the ED drawback mentioned in Section 4.2 (Figure 4.2) is not a critical issue in this research, it could be a critical issue in larger datasets.

Table 7.1: Uniqueness Results.

Dataset	**Cosine**	**ED**	**TS-SS**
20NewsGroup	2.19%	88.13%	**88.92%**
7sector	2.48%	**97.92%**	**97.92%**
WebKB	1.37%	99.50%	**99.51%**
Classic4	1.79%	**98.14%**	**98.14%**
Twitter	0.71%	37.87%	**38.21%**

7.2 Number of Booleans Results

Table 7.2 shows that in overall, Cosine generated around 99% boolean results while the other two metrics did not do so for the same documents. In general the higher percentage of uniqueness and number of boolean values in ED and TS-SS supports the claim that many document pairs which have the same similarity values based on cosine metric, are not exactly same, therefore we believe cosine is not robust enough to distinguish similarities in high level.

Table 7.2: Number of Booleans.

Dataset	Cosine	ED	TS-SS
20NewsGroup	98.71%	0.089%	**0.087%**
7sector	99.64%	**0.20%**	**0.20%**
WebKB	99.61%	**0.12%**	**0.12%**
Classic4	99.87%	**0.44%**	**0.44%**
Twitter	99.47%	**1.00%**	**1.00%**

7.3 Purity Results

Table 7.3 represents the most significant results in comparing the three metrics. As the table shows, ED is the weakest model for clustering. In our biggest dataset, 20NewsGroup, TS-SS outperforms Cosine with a significant difference, while in other datasets TS-SS outperforms Cosine slightly. In fact in the small datasets, there are few types of documents and the chance that documents of the same type get clustered together is higher than the condition where there are several types like 20 types of documents in 20News dataset. Therefore, the significant better result of TS-SS in 20News dataset justifies the robustness and reliability of the model for big data and real world data where the variety of documents/texts are high.

Table 7.3: Purity.

Dataset	Cosine	ED	TS-SS
20NewsGroup	0.46	0.45	**0.86**
7sector	0.73	0.69	**0.75**
WebKB	0.83	0.74	**0.85**
Classic4	0.92	0.80	**0.95**

As we mentioned earlier, due to stochastic outcome of K-Means, we run the algorithm 100 times and selected the best purity result of each measure shown in Table 7.3. In order to show the consistency of the results over 100 runs, the standard deviation of all 100 purities of each dataset for each measure is shown in Table 7.4.

Table 7.4: Standard deviation of purity values.

Dataset	Cosine	ED	TS-SS
20NewsGroup	0.043	0.036	**0.033**
7sector	0.054	**0.027**	0.056
WebKB	0.063	0.091	**0.038**
Classic4	0.097	0.120	**0.086**

However the maximum, minimum, average and standard deviation of purities are represented in Figures 7.1, 7.2, 7.3 and 7.4 respectively.

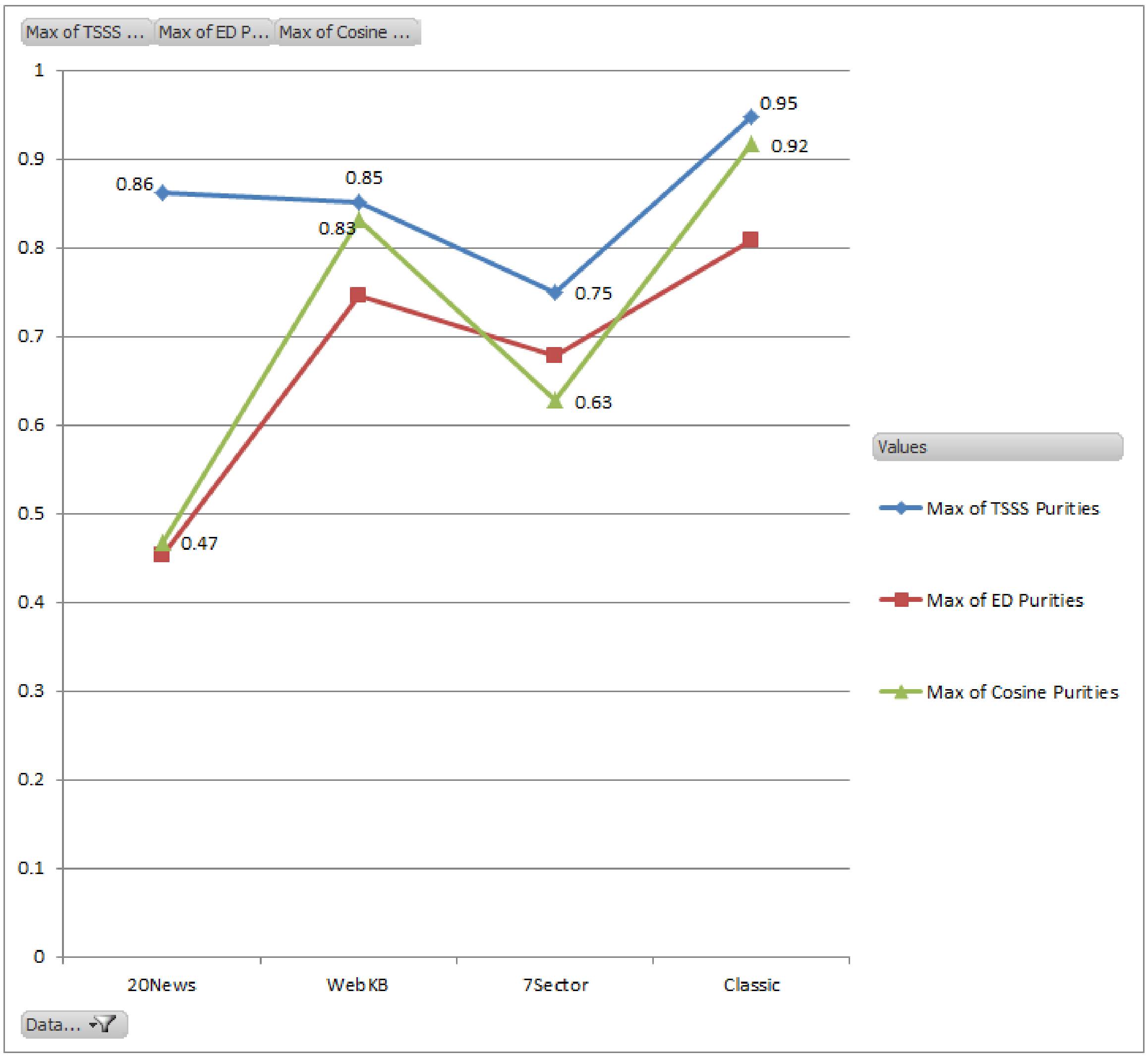


Figure 7.1: Maximum purities.

In each chart, the datasets are sorted from left to right based on the number of categories/classes of documents. Based on what explained in Section 6.1, 20News, WebKB, 7Sector and Classic4 have 20, 10, 7 and 4 categories respectively. Based on the general downward trend in Figures 7.1, 7.2, 7.3 and upward trend in Figure 7.4, we conclude TS-SS gives a better result on data with higher range of classes.

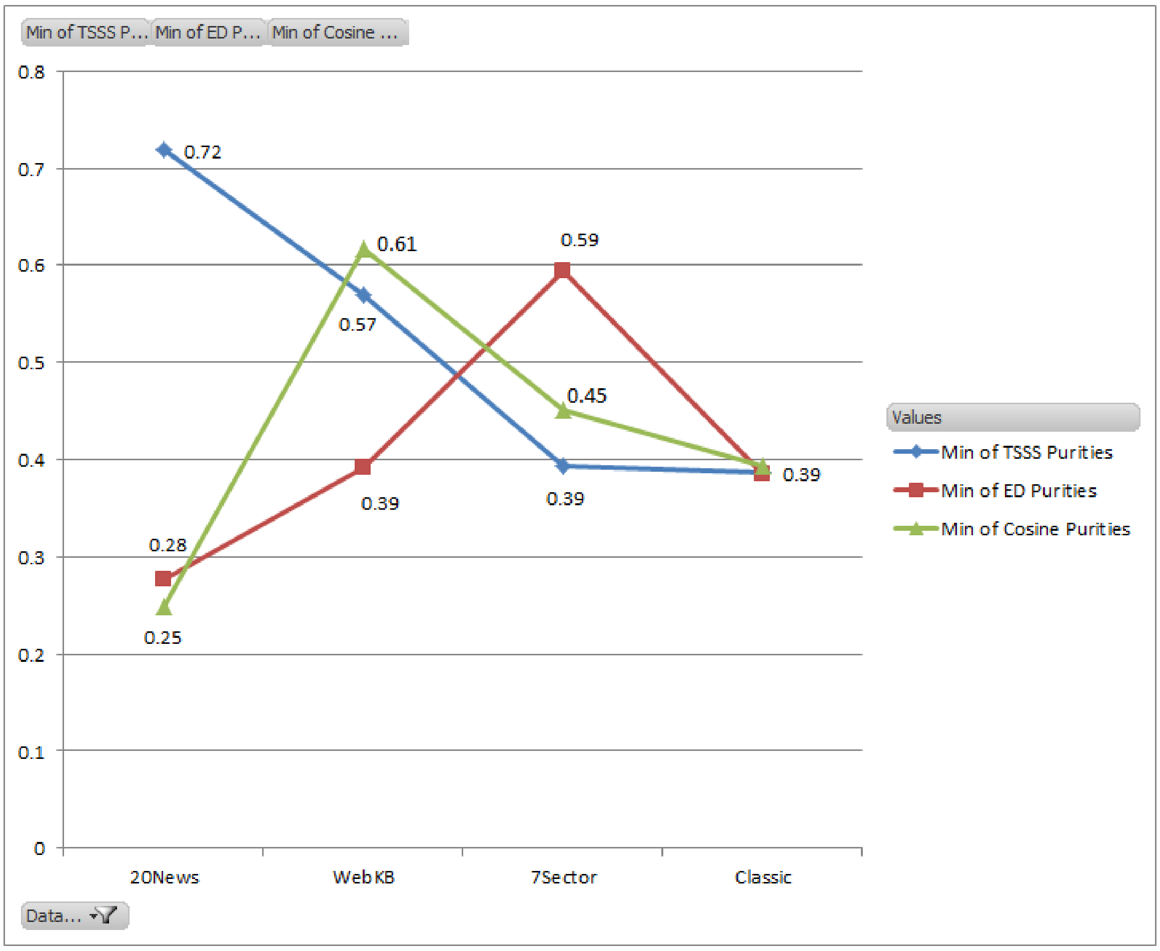


Figure 7.2: Minimum purities.

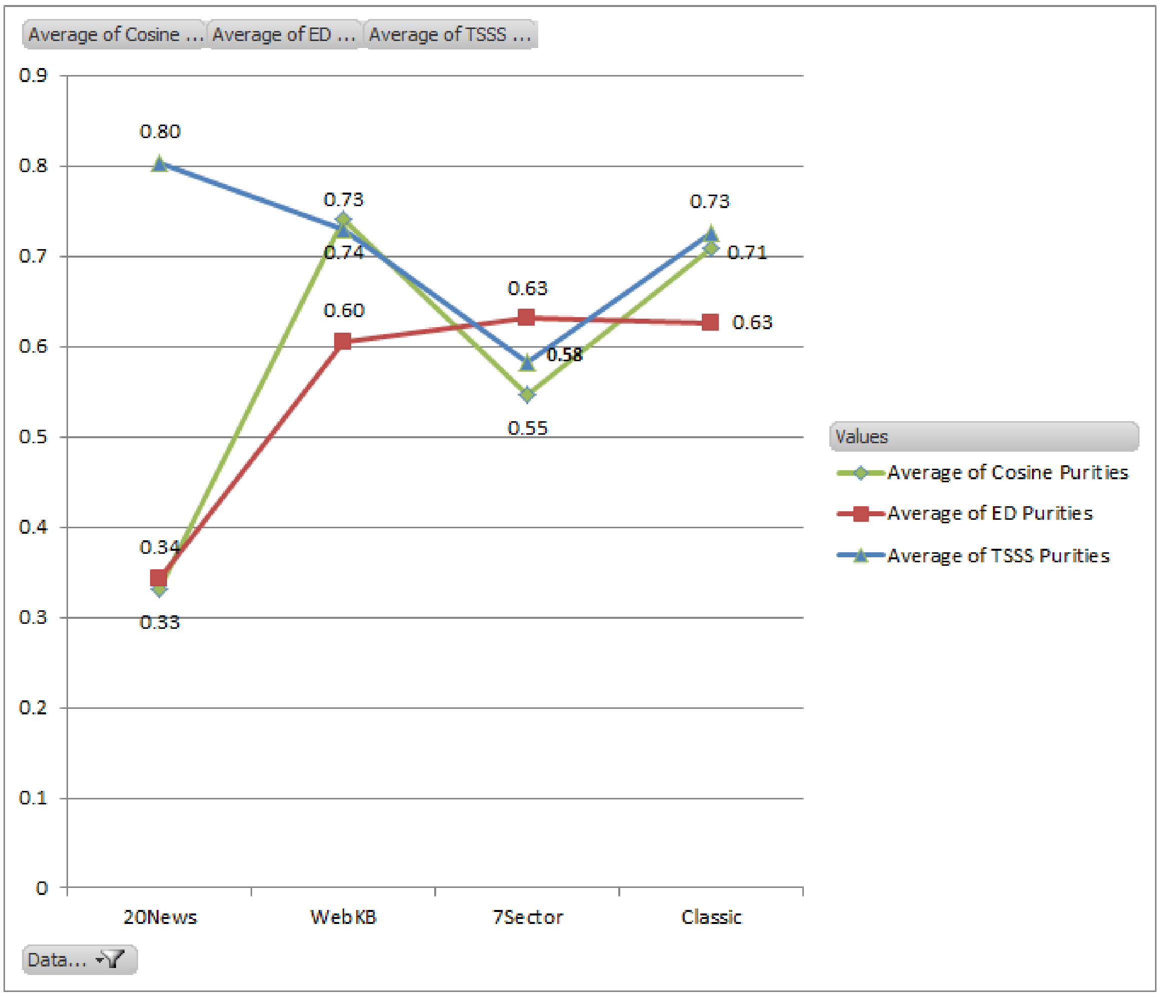


Figure 7.3: Average purities.

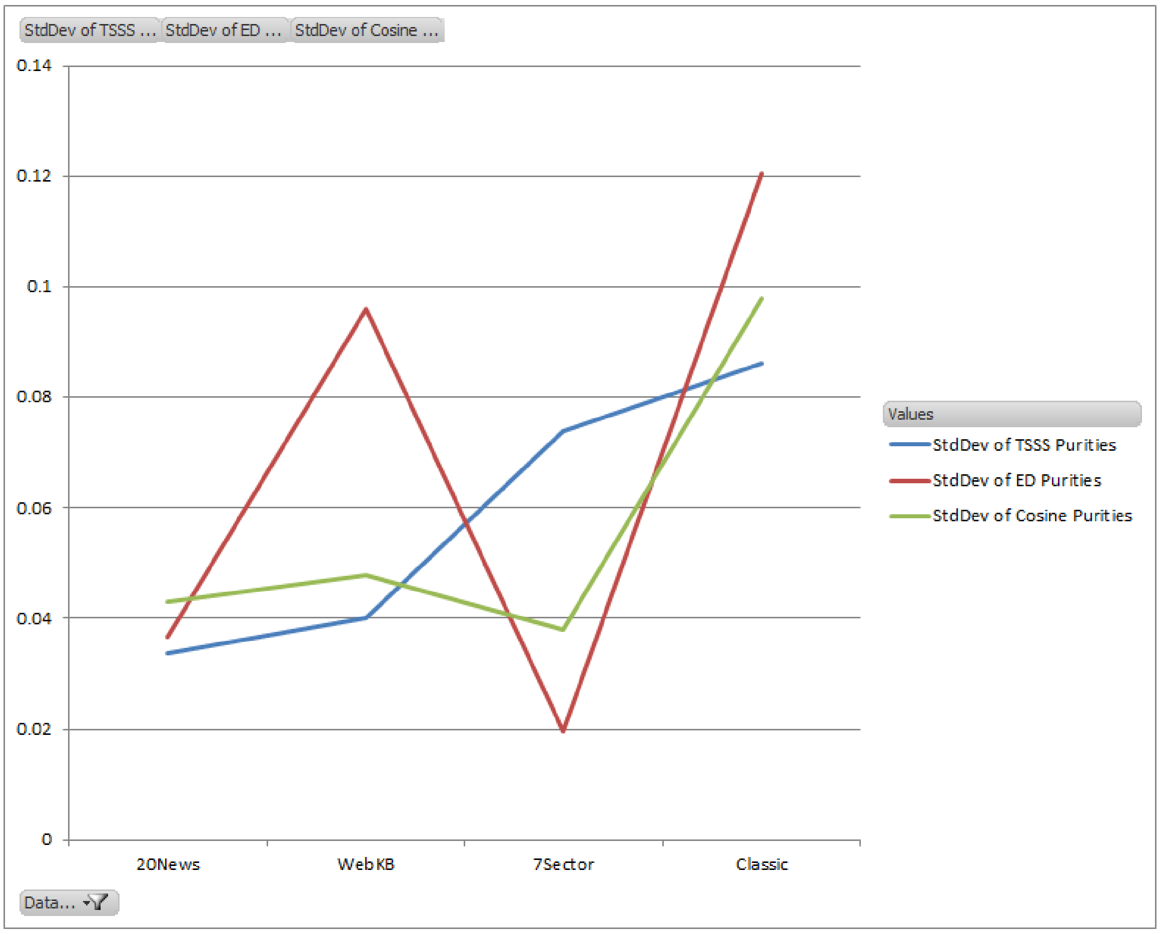


Figure 7.4: Purity's standard deviations.

Although in this thesis the drawbacks in geometric similarity measures are the motivation to come up with a new solution and compare TS-SS with ED and Cosine, the cluster purity results on the same benchmark datasets from other studies (Sachdeva & Kastore 2014, Rafi & Shaikh 2013, Huang 2008) show that TS-SS outperforms Pearson, Jaccard and KLD (Table 7.5). As it can be seen from Table 7.5, Cosine similarity and ED measures in three mentioned studies are different even though they are applied on the same datasets.

Table 7.5: Purity of clusters for geometric and non-geometric similarities from other work.

	Huang (2008)					Rafi&Shaikh (2013)			
	Jaccard	KLD	Pearson	ED	Cosine	Jaccard	KLD	ED	Cosine
20News	0.5	0.38	0.5	0.1	0.5	0.5	0.38	0.23	0.54
Classic	0.98	0.84	0.85	0.56	0.85	0.86	0.89	0.56	0.86
WebKB	0.57	0.75	0.67	0.42	0.68	0.57	0.75	0.45	0.68

The difference could be because of two reasons. First, K-Means is a non-deterministic and stochastic clustering method, therefore it generates different results in several runs. In our study we run K-Means 100 times and then we selected the best purity of all runs for each measure. Huang (2008) runs K-Means for 60 times and Rafi & Sheikh (2013) did not mentioned how many times K-Means have been run. The second reason is, the other two works measured the overall similarity among documents regardless of any specific keyword(s), while in this study, the similarities are measured based on the randomly-generated keywords from top hight TF-IDF scored keywords as explained in Chapter 6. Due to the mentioned reasons, results from the same measures on same datasets from different work are sometimes slightly, and some times significantly different. Recalling the fact the purpose of this study was to cover drawbacks from other geometric measures, though when the results are compared to non-geometric results, the reliability and robustness of TS-SS is still promising and the results are quite competitive and even better in many cases.

7.4 Minimum Gapscore Results

Table 7.6 shows the overall minimum gap score of each measure. Based on our definition in Section 6.2.4, the less gap score means more similarity to the oracle. The result shows the only significant outperformance belongs to TS-SS in 20NewsGroup and in other datasets results are approximately same for all measures.

Table 7.6: Minimum Gapscore.

Dataset	**Cosine**	**ED**	**TS-SS**
20NewsGroup	61876.25	61962.20	**61682.10**
7sector	**6083.30**	6085.70	6085.70
WebKB	**32065.44**	32066.22	32066.22
Classic4	21335.00	21336.00	**21334.70**

7.5 Test Case Results

Referring to the four documents in test case shown in Table 6.3 each document has only one of the keywords in the search query and the common keyword is *"alveolar"*. By looking at proportion of term frequency (TF) to total number of words (W) column in Table 7.7, we notice that document A has the highest similarity to document D and its similarity to B and C is equal, but the numbers in column W, indicate A is slightly more similar to C rather than to B in term of number of words.

Table 7.7: Term frequency (TF), total number of words (W) and their proportion (TF/W) in each document of the test case.

Doc.ID	**W**	**TF of *"alveolar"***	**TF/W**
A	37	2	0.05
B	147	3	0.02
C	119	3	0.02
D	127	6	0.04

By applying Cosine on the four mentioned documents, the similarity value of 1 is returned as the measured similarity value for each pair of documents. This result conveys that all four documents are similar to each other, but no similarity level is measured. Unlike Cosine, Ed and TS-SS measure the similarity levels as expected. As shown in Table 7.8 and Table 7.9, Sim(A,D) > Sim(A,C) > Sim(A,B) where Sim(A,B), Sim(A,C) and Sim(A,D) represents the similarity between A and B, A and C and A and D respectively. This result shows when the mentioned drawbacks about ED does not exist, ED is as powerful as TS-SS in measuring the similarity level, but its low purity results shows it is not as reliable as TS-SS.

Table 7.8: TS-SS similarity values for the test case.

Doc.ID	**A**	**B**	**C**	**D**
A	0	4.00E-5	3.68E-05	3.76E-06
B	4.00E-5	0	3.96E-07	2.26E-05
C	3.68E-05	3.96E-07	0	1.91E-05
D	3.76E-06	2.26E-05	1.91E-05	0

Table 7.9: ED similarity values for the test case.

Doc.ID	**A**	**B**	**C**	**D**
A	0	0.198	0.169	0.039
B	0.198	0	0.029	0.159
C	0.169	0.029	0	0.130
D	0.039	0.159	0.130	0

Chapter 8

Time Complexity

For n documents, there are n^2 relationships. For each relationship the similarity can be measured using α (Cosine, ED and TS-SS). Let $Q = \{t_1, \ldots, t_k\}$ be the set of k terms (keywords) where the searched query is a subset. Measuring similarity for each measure can be done in $O(k)$ and need to process $O(n^2)$ combinations of n documents, so for all relationships the similarity can be computed in $O(n^2\ k)$ running time.

For measuring the uniqueness and number of booleans, we take the entries above the main diagonal only. For n documents there are $\frac{n^2-n}{2}$ elements above the diagonal. Hence counting the unique values and booleans each can be carried out in $O(n^2)$.

The K-Means algorithm clusters n documents in $O(i\ c\ n\ k)$ where i is the number of iteration, c is the number of clusters (number of clusters is equal to number of categories in each similarity matrix) and k is the vector dimension (number of terms). In computing purity values, group the documents of the same cluster is done in $O(n)$ in worse case and finding the dominant category in all clusters is done in $O(c\ \log n)$ if we have more than one cluster, and $O(n)$ if we have only one cluster. Finally calculating purity is done in constant time of $O(c)$. Overall, computing the purity is done in $O(n)$.

For finding the minimum gap score, the sorting algorithm is computed in $O(n^2\ \log n)$ and it dominates the other steps of the algorithm.

Chapter 9

Conclusion

The increasing numbers of textual documents from diverse sources such as different websites (social networks, news, magazines, blogs and medical recommendation websites), publications and articles and medical prescriptions leads to massive amounts of daily complex data. This phenomenon has caused many researchers to focus on analysing the content and measuring the similarities among the documents and texts to cluster them. There are several algorithms used for measuring similarity between documents such as Pearson (Kornbrot 2005), Spearman (Zar 1998), Kullback-Leibler divergence (Kullback & Leibler 1951), Jaccard coefficient (Jaccard 1912), Shannon (Wartena & Brussee 2008), Euclidean Distance (ED) and Cosine similarity (Salton & Buckley 1988). In general the similarity measures can be divided into two categories namely geometric similarities (such as VSM model: ED and Cosine similarity) and non-geometric similarities (such as Pearson, Spearman, Kullback-Leibler divergence, Jaccard coefficient and Shannon). Though the results of many studies (Salton & Buckley 1988, Nelson et al. 2004) show the geometric and VSM based models are more robust in measuring similarities among documents compared to non-geometric models, there are not many work focusing on geometric methods to boost these similarity measures for better results. That is why in this study we focus on geometric methods. One popular method to measure the similarity between documents is to represent the terms within the documents as vectors and measure the similarity among them based on the angle or Euclidean distance between each pair. By only considering these two criteria for similarity measurement, we may miss important underlying semantic similarities in this area. This study gives insights on the drawbacks of geometrical and some non-geometrical similarity measures and provides a novel method to combine the other geometric criteria into a method to measure the similarity level among documents from new prospective. We proposed a new method, TS-SS, to measure the similarity level among documents, in such a way that one hopes to better understand which documents are more (or less) similar. The main purpose of this study is to measure the similarities among documents with high accuracy in order to have a better understanding of which documents are more similar (or less similar). We call this concept *similarity level* which is focused in this study. In this work, *accuracy* referred to the power of a measure in differentiating the similarity level among documents in such a way that one can understand which documents are less, more and most similar. This power of differentiation can be significantly useful for recommendation systems and clusterings.

In Chapter 3 the technique used in this study for text pre-processing steps, Vector Space Model and geometric similarity measures namely Cosine similarity and Euclidean distance, clustering techniques, clustering tools and methods for evaluating clusters explained in details. In fact using text mining, information can be extracted to derive abstract of the words which exist in the documents or to compute that abstracts for the documents based on the words contained in them. After applying text-mining methods to pre-process texts, each document represented as a bag of words, including stemmed words and excluding any punctuation, stop words, linking words, new lines, URLs and other noisy data explained in details.

Then using VSM, each document represented as a vector for any desired searched query (in this study each search query contains three keywords). VSM is an algebraic model widely used in information retrieval and data mining. The model uses natural language processing techniques to represent each documents/texts as set of vectors with addition and scalar multiplication which introduced by Salton (Salton & Buckley 1988). Each vector describes an object (documents and corpus of texts) using n-dimensional vectors which each dimension representing the frequency of a certain term (keyword) in a document using a term weighting model called as TF-IDF model.

After representing documents/texts as vectors using VSM, two traditional similarity measures, namely Cosine similarity and Euclidean distance explained, which are widely used to identify the similar documents. Cosine similarity computes the pairwise similarity between two documents using dot product and magnitude of vector document A and vector document B in high-dimensional space. This measure helps to avoid the bias caused by documents' length as it uses the angle between two vectors to compute the similarity and has nothing to do with the length of the vectors. The direction of vectors play the crucial role when the similarities are measured using Cosine method. Euclidean distance (ED) is another geometrical measure used to measure similarity of two documents. Each document is represented as a point in space based on term frequency of n terms (representing n dimension). ED computes the difference between two points in n-dimensional space based on their coordinates.

The next step after measuring similarities, is clustering. Clustering divides data into groups (clusters) that are meaningful, useful, or both. Using clustering, data objects can be grouped based on information which are found in the data that describes the objects and/or the relationships among them. The purpose is to group a similar objects (documents) which are related to each other and are different from the other objects in other groups. The more similarity within a group, the more differences between groups and more accurate and distinct clustering (Tan et al. 2013). There are enormous number of techniques and algorithms for document and text clustering, but in general two main approaches are agglomerative hierarchical clustering and K-

Means (Steinbach et al. 2000). Based on comprehensive comparisons in other studies (Tan et al. 2013), K-Means performs better clustering, that is why K-Means is selected for clustering in this study. K-Means is supervised learning algorithm that is widely used for clustering plotted data points in space (MacQueen et al. 1967). The algorithm clusters data points into K clusters based on K given centroids. In order to get the most discriminated clusters, centroids should be placed as far as possible from each other. Then each point belongs to the nearest centroid to form a cluster. When all nodes are assigned to a centroid, we need to re-calculate K new centroids as new centre of the clusters resulting from the previous step. After K new centroids identified, new assigning should be done to the same data set points to the nearest new centroid. The tool used for clustering is WEKA.

After applying K-Means clustering, evaluating the quality of the clusters is the first way to examine the reliability of the method. Clustering validation can be done via Internal validation and External validation. Internal validations is used when there is no priori information available about the data and it can be done based on the clustering structure. Some internal validity measures and indices are Cohesion and Separation, Silhouette Coefficient, Bic index, Calinski-Harabasz index, Davies-Bouldin index (DB), Dunn index and Niva index (Rendón et al. 2011). External validation indexes is suitable for situations where we have a priori knowledge of dataset information. In this study we applied the external clustering method, because we did the experiments on the labelled datasets which can be used to evaluate the quality of clusters. Some external validity indices are F-measure, Purity and Entropy. We used Purity method. Purity is a simple evaluation measure which is very similar to Entropy. To compute the purity, first each cluster is assigned to the class which is the most frequent in the cluster. Then in order to compute the accuracy of the assignment, we divide the number of correctly assigned documents by total number of documents in the cluster, in ideal situation, if there are certain number of documents in a cluster and all of the documents are from the same class, the purity is 1 which is the highest cluster quality.

In Chapter 4 drawbacks of Cosine similarity and ED have been scrutinized from different views and it is explained why existing geometric measures are not robust enough to measure the similarity level accurately. Cosine returns the same similarity result among pairs of overlapped vectors with different magnitudes. Cosine measures similarity based on angular difference between vectors and it ignores the magnitude of vectors. In fact the higher proportion of the terms in document's texts, leads to have a longer magnitudes and signifies particular terms from a searched query have higher importance in particular documents (Figure 4.1). That is why Cosine is not a reliable measure in recognizing which documents are more or less similar (similarity level). Although ED does not ignore the magnitude of vectors, the results of this research and other research show this measure is one of the weakest models amongst similarity measures. The main issue with ED is that it does not take angular difference between vectors. Three vector of P,Q and R with equal magnitudes can be drawn with same same ED from vector R while P might have smallest angular difference and R might have biggest angular difference to M (Figure 4.2). In this case ED(M,P)=ED(M,Q)=ED(M,R) while angular difference shows P is more similar to M.

In Chapter 5 a new method called TS-SS proposed, which covers the mentioned drawbacks to measure the similarity level among documents. The measure is a product of Triangle's Area Similarity (TS) and Sector's Area Similarity (SS) which takes angular difference and magnitude difference into account. TS and SS complete each other and that is the reason we combine them by multiplying them together. The range of TS-SS measure is from 0 to ∞. The reason for choosing multiplication but not summation to combine TS and SS is that in some cases the value of TS and SS are disproportionate where one is extremely larger then the other one. For example, in Figure 4.1, TS similarity is too big (TS(A,B)=5.91) while SS similarity is too small (SS(A,B)=0.047). If we use summation, they can not effect on each other significantly to give the realistic similarity value and we get TS(A,B)+SS(A,B)=5.95 $>$ TS(A,D)+SS(A,D)=2.96 which is a false result because A is more similar to B as discussed earlier. But if we use multiplication we get TS(A,B)$\cdot$ SS(A,B)=0.27 $<$ TS(A,D)$\cdot$SS(A,D)=1.71 which is a true result. Using TS-SS method, similarity of 0 happens only when ED=MD=0 and it shows two vectors are absolutely identical in term of direction and magnitude which indicates the maximum similarity between two documents.

In Section 6 five datasets used for experiments and four evaluation methods namely Uniqueness, Number of Booleans, Purity and Minimum Gapscore are described in detail which used for making comparison among the traditional measures with TS-SS to show how the mentioned drawbacks about Cosine and ED may effect the result of similarity level and clustering quality. The *Uniqueness* test computes the percentage of unique values in each similarity matrix. The outcome of this evaluation indicates the existence possibility of drawbacks mentioned in Chapter 4. In fact it shows how accurate a measure can distinguish the differences among similarities. If the number of unique similarities are higher, means more variation among similarities are detected. The *Number of Booleans* test counts the number of boolean values in the similarity matrices generated by each measure and shows how many percent of values are booleans. The main purpose of this evaluation is to show which measure produces more boolean values and gives less variation. In fact this is another attempt to evaluate the highlighted drawbacks in Chapter 4. *Purity* is widely used for measuring the quality of clusters based on the label of documents in each cluster. In this study labels identify which category each document belongs. As described in details in Section 3.6.2, to measure the purity of clusters, each cluster is assigned to the category which is the most frequent in the cluster. Then the assignment accuracy of cluster C_i which contains n_i documents is measured by the following formula (*Evaluation of clustering @ONLINE* 2009). *Minimum Gapscore* test compares the similarity matrices with their associated oracle. As the documents are labelled, we are able to construct the oracle matrix using the existing document labels from the datasets. The less gap score means more similarity between measured values and oracle.

In Chapter 7 the proposed method and other similarity measures are applied on five datasets to cluster the documents and then and the results of clustering from the proposed method and other similarity measures are compared using four evaluation methods. The evaluations' results show the proposed model, TS-SS outperforms the other measures as following. The results of *Uniqueness* test shows Cosine similarity has the very low percentage of uniqueness and it conveys that the drawbacks mentioned about this model in Section 4.1 (Figure 4.1) is a significant issue. This low accuracy in terms of recognizing the differences among similarity values causes the lack of distinguishment between documents with higher similarities and documents with lower similarities. Although intangible difference between ED uniqueness and TS-SS uniqueness indicates that the ED drawback mentioned in Section 4.2 (Figure 4.2) is not a critical issue in this research, it could be a critical issue in larger datasets.

The result of *Number of Booleans* shows that in overall, Cosine generated around 99% boolean results while the other two metrics did not do so for the same documents. In general the higher percentage of uniqueness and number of boolean values in ED and TS-SS supports the claim that many document pairs which have the same similarity values based on cosine metric, are not exactly same, therefore we believe cosine is not robust enough to distinguish similarities in high level.

The result of *Purity* shows ED is the weakest model for clustering. In our biggest dataset, 20NewsGroup, TS-SS outperforms Cosine with a significant difference, while in other datasets TS-SS outperforms Cosine slightly. In fact in the small datasets, there are few types of documents and the chance that documents of the same type get clustered together is higher than the condition where there are several types like 20 types of documents in 20News dataset. Therefore, the significant better result of TS-SS in 20News dataset justifies the robustness and reliability of the model for big data and real world data where the variety of documents/texts are high. The results from *Minimum Gapscore* test shows the only significant outperformance belongs to TS-SS in 20NewsGroup (which is the biggest dataset with highest variety of categories) and in other datasets results are approximately same for all measures.

Finally in Chapter 8 the time complexity of Cosine similarity, Euclidean distance, TS-SS and K-Means algorithm which is used for clustering are calculated. The results show three measures namely Cosine Similarity, ED and TS-SS have the same complexity.

Bibliography

Attribute-Relation File Format (ARFF) (2008), http://www.cs.waikato.ac.nz/ml/weka/arff.html.

Balabanović, M. & Shoham, Y. (1997), 'Fab: content-based, collaborative recommendation', *Communications of the ACM* pp. 66–72.

Becker, J. & Kuropka, D. (2003), Topic-based vector space model, *in* 'Proceedings of the 6th International Conference on Business Information Systems', pp. 7–12.

Bigi, B. (2003), *Using Kullback-Leibler distance for text categorization*, Springer.

Bilenko, M. & Mooney, R. J. (2003), Adaptive duplicate detection using learnable string similarity measures, *in* 'Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 39–48.

Bo, Y. & Luo, Q. (2007), Personalized web information recommendation algorithm based on support vector machine, *in* 'Intelligent Pervasive Computing, 2007. IPC. The 2007 International Conference on', IEEE, pp. 487–490.

Cha, S.-H. (2007), 'Comprehensive survey on distance/similarity measures between probability density functions', *International Journal Of Mathematical Models And Methods In Applied Science* .

Chim, H. & Deng, X. (2008), 'Efficient phrase-based document similarity for clustering', *Knowledge and Data Engineering, IEEE Transactions* pp. 1217–1229.

Classic3 and Classic4 DataSets (2010).
URL: http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/

CMU World Wide Knowledge Base Project (2011).
URL: http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/

Das, S., Egecioglu, Ö. & El Abbadi, A. (2009), 'Anonymizing edge-weighted social network graphs', *Computer Science, UC Santa Barbara, Tech. Rep. CS-2009-03* .

Evaluation of clustering @ONLINE (2009).
URL: http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

Hammouda, K. M. & Kamel, M. S. (2002), Phrase-based document similarity based on an index graph model, *in* 'Proceedings of the 2002 IEEE International Conference on Data mining', IEEE, pp. 203–210.

Han, X., Wang, L., Crespi, N., Park, S. & Cuevas, Á. (2015), 'Alike people, alike interests? inferring interest similarity in online social networks', *Decision Support Systems* pp. 92–106.

Hannon, J., McCarthy, K. & Smyth, B. (2011), Finding useful users on twitter: twittomender the followee recommender, *in* 'Advances in Information Retrieval', Springer, pp. 784–787.

Hearst, M. (2003), 'What is text mining', *SIMS, UC Berkeley* .

Heidarian, A. (2011), Multi-clustering users in twitter dataset, *in* 'International Conference on Software Technology and Engineering, 3rd (ICSTE 2011)', ASME Press.

Hsieh, S.-M., Huang, S.-J., Hsu, C.-C. & Chang, H.-C. (2004), Personal document recommendation system based on data mining techniques, *in* 'Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on', IEEE, pp. 51–57.

Huang, A. (2008), Similarity measures for text document clustering, *in* 'Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand', pp. 49–56.

Jaccard, P. (1912), 'The distribution of the flora in the alpine zone. 1', *New phytologist* pp. 37–50.

Karman, S. S. & Ramaraj, N. (2008), 'Similarity-based techniques for text document classification', *Int. J. SoftComput* pp. 58–62.

K-Means Clustering (2015), http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html. Accessed: 2015-05-02.

Kornbrot, D. (2005), 'Pearson product moment correlation', *Encyclopedia of Statistics in Behavioral Science* .

Kullback, S. & Leibler, R. A. (1951), 'On information and sufficiency', *The Annals of Mathematical Statistics* pp. 79–86.

Lai, C.-H. & Liu, D.-R. (2009), 'Integrating knowledge flow mining and collaborative filtering to support document recommendation', *Journal of Systems and Software* pp. 2023–2037.

Lebanon, G. (2006), 'Metric learning for text documents', *Pattern Analysis and Machine Intelligence, IEEE Transactions* pp. 497–508.

Liao, H.-Y., Chen, K.-Y. & Liu, D.-R. (2015), 'Virtual friend recommendations in virtual worlds', *Decision Support Systems* pp. 59–69.

MacQueen, J. et al. (1967), Some methods for classification and analysis of multivariate observations, *in* 'Proceedings of the fifth Berkeley symposium on mathematical statistics and probability', Oakland, CA, USA., pp. 281–297.

Manning, C. D., Raghavan, P., Schütze, H. et al. (2008), *Introduction to information retrieval*, Vol. 1, Cambridge university press Cambridge.

Mao, W. & Chu, W. W. (2002), Free-text medical document retrieval via phrase-based vector space model., *in* 'Proceedings of the AMIA Symposium', American Medical Informatics Association, p. 489.

Mihalcea, R., Corley, C. & Strapparava, C. (2006), Corpus-based and knowledge-based measures of text semantic similarity, *in* 'AAAI', pp. 775–780.

Nayak, P. & Raghavan, P. (2014), 'Scoring, term weighting and the vector space model'. Lecture notes of Information Retrieval and Web Search course, Stanford University.

Nelson, M. L., Bollen, J., Calhoun, J. R. & Mackey, C. E. (2004), User evaluation of the nasa technical report server recommendation service, *in* 'Proceedings of the 6th annual ACM international workshop on Web information and data management', ACM, pp. 144–151.

Pazzani, M. J., Muramatsu, J., Billsus, D. et al. (1996), Syskill & webert: Identifying interesting web sites, *in* 'AAAI/IAAI, Vol. 1', pp. 54–61.

Rafi, M. & Shaikh, M. S. (2013), 'An improved semantic similarity measure for document clustering based on topic maps', *arXiv:1303.4087* .

Rendón, E., Abundez, I., Arizmendi, A. & Quiroz, E. (2011), 'Internal versus external cluster validation indexes', *International Journal of computers and communications* pp. 27–34.

Sachdeva, S. & Kastore, B. (2014), Document clustering: Similarity measures, Technical report, Indian Institute of Technology Kanpur.

Sahami, M. & Heilman, T. D. (2006), A web-based kernel function for measuring the similarity of short text snippets, *in* 'Proceedings of the 15th international conference on World Wide Web', AcM, pp. 377–386.

Salton, G. & Buckley, C. (1988), 'Term-weighting approaches in automatic text retrieval', *Information processing & management* pp. 513–523.

Sammut, C. & Webb, G. I. (2011), *Encyclopedia of machine learning*, Springer Science & Business Media.

Silva, N. B., Tsang, I.-R., Cavalcanti, G. D. & Tsang, I.-J. (2010), A graph-based friend recommendation system using genetic algorithm, *in* 'Evolutionary Computation (CEC), 2010 IEEE Congress on', IEEE, pp. 1–7.

Singhal, A., Buckley, C. & Mitra, M. (1996), Pivoted document length normalization, *in* 'Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 21–29.

Soucy, P. & Mineau, G. W. (2005), Beyond tfidf weighting for text categorization in the vector space model, *in* 'IJCAI', pp. 1130–1135.

Steinbach, M., Karypis, G., Kumar, V. et al. (2000), A comparison of document clustering techniques, *in* 'KDD workshop on text mining', Boston, pp. 525–526.

Stemming (2015), https://en.wikipedia.org/wiki/Stemming. Accessed: 2015-06-05.

Strehl, A., Ghosh, J. & Mooney, R. (2000), Impact of similarity measures on web-page clustering, *in* 'Workshop on Artificial Intelligence for Web Search (AAAI 2000)', pp. 58–64.

Tan, P.-N., Steinbach, M. & Kumar, V. (2013), 'Data mining cluster analysis: Basic concepts and algorithms'.

Text mining, also known as intelligent text analysis, text data mining or knowledge-discovery in text (KDT) (2014), http://www.mu-sigma.com/analytics/thought_leadership/cafe-cerebral-text-mining.html.

Text Mining (Big Data, Unstructured Data) (2015), http://www.statsoft.com/Textbook/Text-Mining. Accessed: 2015-05-10.

The 20 Newsgroups data set (2008).
URL: http://qwone.com/~jason/20Newsgroups/

Turney, P. (2001), 'Mining the web for synonyms: Pmi-ir versus lsa on toefl'.

Wartena, C. & Brussee, R. (2008), Topic detection by clustering keywords, *in* '19th International Workshop on Database and Expert Systems Application, 2008. DEXA'08.', IEEE, pp. 54–58.

WebKB (2010).
URL: http://www.csmining.org/index.php/webkb.html

Wong, S. M., Ziarko, W. & Wong, P. C. (1985), Generalized vector spaces model in information retrieval, *in* 'Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 18–25.

Wu, W., Zhang, B. & Ostendorf, M. (2010), Automatic generation of personalized annotation tags for twitter users, *in* 'Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics', Association for Computational Linguistics, pp. 689–692.

Xie, X. (2010), Potential friend recommendation in online social network, *in* 'Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)', IEEE, pp. 831–835.

Yang, J. & Leskovec, J. (2011), Patterns of temporal variation in online media, *in* 'Proceedings of the fourth ACM international conference on Web search and data mining', ACM, pp. 177–186.

Zar, J. H. (1998), 'Spearman rank correlation', *Encyclopedia of Biostatistics* .

Zhang, T., Tang, Y. Y., Fang, B. & Xiang, Y. (2012), 'Document clustering in correlation similarity measure space', *Knowledge and Data Engineering, IEEE Transactions* pp. 406–410.