



**INVESTIGATION OF USING THE LSA MODEL WITH
SIMILARITY METRICS FOR SEMANTIC-BASED WEB
DOCUMENT CLUSTERING**

Mashhood ALI

Master's Thesis

Department: Software Engineering

Supervisor: Assist. Prof. Dr. Aytuğ BOYACI
JANUARY-2018

REPUBLIC OF TURKEY
FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

INVESTIGATION OF USING THE LSA MODEL WITH SIMILARITY METRICS FOR
SEMANTIC-BASED WEB DOCUMENT CLUSTERING

MASTER'S THESIS
MASHHOOD ALI ALI
151137118

Submission Date to the Institute: January 2018

Thesis Presentation Date: 19 January 2018

Thesis Supervisor: Assist. Prof. Dr. Aytuğ BOYACI (F.Ü)



Other Jury Members: Assoc. Prof. Dr. Murat KARABATAK (F.Ü)



Assist. Prof. Dr. Ahmet Arif AYDIN (İ.Ü)



JANUARY-2018

ACKNOWLEDGEMENT

I would first like to thank my supervisor, Asst. Prof. Dr. Aytuğ BOYACI and my co-supervisor, Asst. Prof. Dr. Mustafa ULAŞ for their great help and guidance, and through the process of researching and writing this documentation, which it possible to accomplish this work.

I would also express a deep sense of gratitude to my parents and to my friends for providing me with continuous encouragement and their constant support throughout my years of study. This accomplishment would never have been happened without their support. Thank you.

Mashhood Ali
ELAZIĞ - 2018

TABLE OF CONTENTS

	<u>Page No.</u>
ACKNOWLEDGEMENT	II
TABLE OF CONTENTS	III
ÖZET	VI
ABSTRACT	VII
LIST OF FIGURES.....	VIII
LIST OF TABLES.....	IX
LIST OF ABBREVIATIONS	X
1. INTRODUCTION.....	1
1.1. Preface.....	1
1.2. Aim of the Study	2
2. LITERATURE REVIEW AND BACKGROUND.....	3
2.1. Literature Review	3
2.2. Data clustering	6
2.2.1. Partitional Clustering Algorithms	9
2.2.2. Hierarchical Clustering Algorithms	9
2.2.2.1. Agglomerative Hierarchical Clustering (AHC)	9
2.2.2.2. Divisive Hierarchical Clustering (DHC).....	9
2.3. Web Document Clustering.....	10
2.4. Natural Language Processing.....	10
2.5. Term-Document Text Representation.....	12
2.6. LSA Model.....	15
2.7. Text Semantic Similarity Measurements	19
2.8. K-Means Algorithm	23
3. TEXT MINING	26
3.1. Steps in Text Mining.....	27

3.2.	Principal Fundamental Text Mining Techniques	28
3.2.1.	Information Extraction	28
3.2.2.	Categorization	28
3.2.3.	Clustering	29
3.2.4.	Information Visualization	29
3.3.	Text Mining Applications	29
3.3.1.	Academic and Research Field	29
3.3.2.	Social Media	30
3.3.3.	Opinion Mining and Sentiment Analysis	30
3.3.4.	Biomedical Text Mining	30
3.3.5.	Anti-Spam Filtering of Emails	31
3.3.6.	Digital Libraries	31
3.3.7.	Business Intelligence.....	32
3.3.8.	Software Environment	32
3.4.	Semantic-Based Text Mining.....	33
3.5.	LSA-Based Text Document Clustering	33
4.	PROBLEM STATEMENT	35
4.1.	Semantic-Based Web Document Clustering	35
4.2.	LSA-Based Web Document Clustering	35
5.	PROPOSED APPROACH AND METHODOLOGY	37
5.1.	Proposed Approach Steps	37
5.1.1.	Natural Language Processing (NLP) Pipeline	37
5.1.2.	Term-Document Matrix (TDM Matrix).....	38
5.1.3.	LSA Model.....	38
5.1.4.	Investigated Semantic Similarity Measurements	38
5.1.5.	K-Means Algorithm	38
5.2.	Cluster Evaluation Methodology	39
5.3.	Experimental Settings	40
5.3.1.	BBC Dataset.....	41
5.3.2.	CMU World Wide Knowledge Base (Web->KB) Project.....	41

6.	RESULTS	42
7.	DISCUSSIONS AND CONCLUSIONS.....	45
	REFERENCES.....	46
	CURRICULUM VITAE.....	53



ÖZET

SEMANTİK BAZLI WEB DOKÜMANI KÜMELENMESİ İÇİN BENZERİ METRİKLİ LSA MODELİNİN KULLANIMININ İNCELENMESİ

Web belge kümelemesi, benzer web belgelerini, aynı kümedeki belgelerin diğer kümelerdeki belgelere göre semantik olarak daha yakın kategorize edildiği gruplar halinde bir araya getirmek için veri kümeleme tekniklerini kullanmaktadır. Belgeleri kümeleme yöntemlerinden biri, bu belgelerin içerdikleri konulara göre gruplandırılmasına dayanmaktadır. Konu tabanlı web belge kümeleme yönteminde kullanılan temel teknik, veri setinde bulunan terimler ve belgeler gibi her öge için veri seti düzeyinde bir semantik (ör. konular) türeten ve LSA (Latent Semantic Analysis) olarak bilinen semantik analiz modelidir. LSA modeli literatürde, farklı şekillerde, varyasyonlarda ve farklı amaçlarda kullanılmıştır.

Mevcut durumda LSA modelinin birçok kullanımı bulunduğundan, bu çalışmada, metin dokümanlarını semantik olarak kümelemede LSA modelinin en iyi şekilde kullanımı incelenmiştir. Bu sebeple, web belgelerinin kümelenmesinde en iyi performansı gösteren varyasyonu bulmak amacıyla LSA modelinin altı farklı semantik-benzerlik ölçümü ile kombinasyonları incelenmiştir. Metin kümelemesinde LSA modelini kullanımının en iyi varyasyonu, yine bu varyasyonun en çok kullanılan iki web dokümanı veri setine uygulanmasından sonra bulunmuştur. Sonuçlar aynı zamanda, web belge kümelemesi için LSA modelinin kullanımındaki her varyasyonun performansını göstermektedir.

Anahtar Kelimeler: Web Belge Kümeleme, LSA Modeli, Metin Madenciliği, Semantik Benzerlik Ölçümleri.

ABSTRACT

INVESTIGATION OF USING THE LSA MODEL WITH SIMILARITY METRICS FOR SEMANTIC-BASED WEB DOCUMENT CLUSTERING

Web document clustering uses data clustering techniques to group similar web documents into groups, where the documents from the same cluster are more semantically similar than the documents in the other clusters. One of the methods of clustering the documents is based on the topics they contain. The main technique used for topic-based web document clustering is the using of a semantic-analysis model called Latent Semantic Analysis (LSA), which derives a corpus-level semantics (i.e. topics) for every element in the corpus such as, terms and documents. The LSA model has been used in the literature in different ways, variations and for different applications.

In this study, we experimentally investigate the best use of the LSA model in semantically clustering the text documents, as there is more than one possible variation when one uses and implements the LSA model. To do so, we examined the LSA model in different combinations with six different semantic-similarity measures to find the best possible variation, which performs best in clustering web documents. The best variation of using the LSA model in text clustering was found after applying it to two commonly used web document datasets. The results also demonstrate the performance of each variation of using LSA model for the task of web document clustering.

Keywords: Web Document Clustering, LSA Model, Text Mining, Semantic Similarity Measures.

LIST OF FIGURES

	<u>Page No.</u>
Figure 2.1. General view of data clustering.....	7
Figure 2.2. Main steps in data clustering	8
Figure 2.3. The taxonomy of data clustering approaches	8
Figure 2.4. LSA model	16
Figure 2.5. Intersection and union of two finite sets	20
Figure 2.6. The concept of cosine similarity	20
Figure 2.7. Scatter diagrams for different correlation coefficient values	22
Figure 2.8. K-means example	24
Figure 3.1. Text mining dimensions	26
Figure 3.2. Basic process of text mining	27
Figure 5.1. Steps of the proposed approach.....	37
Figure 5.2. Purity as an external evaluation criterion for cluster quality	39
Figure 6.1. Average purity results for both datasets S.....	42
Figure 6.2. Average entropy results for both datasets	43

LIST OF TABLES

	<u>Page No.</u>
Table 2.1. Term-document matrix for the assumed dataset.....	13
Table 2.2. TFIDF weighted matrix for the assumed dataset.....	15
Table 2.3. Term-document matrix	16
Table 2.4. SVD term matrix	16
Table 2.5. Singular value matrix.....	17
Table 2.6. SVD document matrix	17
Table 2.7. Reduced singular value matrix	17
Table 2.8. Reduced term-document matrix	18
Table 6.1. Average purity results for both datasets	42
Table 6.2. Average entropy results for both datasets.....	43

LIST OF ABBREVIATIONS

AHC	: Agglomerative Hierarchical Clustering
BBC	: British Broadcasting Corporation
BNC	: British National Corpus
CMU (Web->KB)	: Carnegie Mellon University (World Wide Knowledge Base)
CoreNLP	: CoreNatural Language Processing
DHC	: Divisive Hierarchical Clustering
DT	: Determiner
GAL	: Genetic Algorithm method based on a Latent semantic model
GATE	: General Architecture for Text Engineering
GO	: Gene Ontology
GOClonto	: Gene Ontology Clonto
HTML	: Hypertext Markup Language
IBM	: International Business Machines
IEEE	: Institute of Electrical and Electronics Engineers
IN	: Preposition or subordinating conjunction
JJ	: Adjective
LEDA	: Library of Efficient Data Types and Algorithms
LSA	: Latent Semantic Analysis
LSI	: Latent Semantic Index
MLDC	: Multilingual Document Clustering
NLP	: Natural Language Processing
NN	: Noun, singular or mass
PCC	: Pearson Correlation Coefficient
PDF	: Portable Document Format
PLSA	: Probabilistic Latent Semantic Analysis
POS	: Part of Speech Tagging
PRP	: Personal pronoun
PSO	: Particle Swarm Optimization algorithm

PubMed	: Publication Medical
SVD	: Singular Value Decomposition
SVM	: Support Vector Machine
TDM	: Term-Document Matrix
TF/IDF	: Term Frequency / Inverse Document Frequency
VBG	: Verb, gerund, or present participle
VBP	: Verb, non-3rd person singular present
XML	: Extensible Markup Language



1. INTRODUCTION

This chapter introduces our work of investigating the usage of the Latent Semantic Analysis (LSA) model with different semantic similarity metrics for the text-mining task of semantic-based web document clustering. Then, this chapter includes the preface, which briefly presents our research problem and hypothesis. Then it describes the aim and purpose of the study by describing its importance in the field of text mining.

1.1. Preface

From the starting point of the invention of technology, the size of data has been growing vastly. In general, the data has various dimensions and properties, which makes its processing and analysis more complicated and costly in terms of size and time complexity. There are many sources and applications for generating a variety of forms of textual data such as bioinformatics, digital imaging, economics, social media, and many other resources that have produced and are still generating many high-dimensional and high-volume data sets. Although having different sources of data generation, unstructured text remains the main data format on the web. Therefore, having efficient and intelligent algorithms for processing and analyzing textual data is an avoidable need in most applications, which have the textual data as their main format of data [1].

Textual data comes in the human language, English in our case. Thus, there is a need to explore its linguistic properties to extract the knowledge from it. In the era of data analysis, we need efficient and accurate systems, which can handle the large amount of textual data for extracting insights and knowledge from it. Text clustering is a unavoidable phase for information retrieval applications, which helps documents being topically clustered and ready for being matched for incoming queries, such as in web search engines.

On the other hand, text classification is known as a supervised machine learning problem where the classes are already being known to the system earlier and are set in advance for each training document. Unlike classification, document clustering is an unsupervised machine learning problem, where there are no classes that are predefined to system, yet it can group most related documents in the same cluster and different documents in different clusters, based on some similarity measurements. For instance, document clustering benefits

are widely used in the field of information retrieval, which creates similarity-links between related/similar documents. Thus, it makes more accurate and easier retrieving of related documents compared to the received query [2, 3].

In this work we handle the text-mining problem of semantic-based web document clustering. However, before going in depth into our research problem, some preliminary knowledge is recommended for the reader, as in the next two chapters.

1.2. Aim of the Study

This study intends to find the best possible variation of using the LSA model with semantic similarity measurements for the task of web document clustering. Web document clustering is one of the main modules in the current web search engines, which enables the web documents to cluster and be ready for the user queries to produce more accurate and fast results for its users. Semantic based web document clustering enables the web search to produce semantically more relevant results for the user queries not only for textual data, but also for retrieving images and videos [4, 5].

2. LITERATURE REVIEW AND BACKGROUND

The previous chapter stated our research introduction. Before moving forward to our proposed approach, this chapter presents the related theoretical background knowledge, which can help the reader to better understand our proposed approach and our experimental setups and results. In this chapter data clustering is defined with its major versions. Then the standard pipeline needed for LSA based web document clustering is described, including the following steps; natural language processing (NLP) steps, term-document text representation, LSA model of semantic learning, the common text semantic-similarity measurements, and finally, the main data-clustering algorithm of K-Means.

2.1. Literature Review

The literature is considered as a major source of information probably for all scientific researchers during their work to obtain sufficient knowledge and experiences from previous experiments, also it helps them to extract new ideas even for their laboratory studies [2]. The latent semantic model, also known as latent semantic indexing, has been used widely for performing text document clustering [6-9]. Due to different possible variations of using the LSA model, some work has been done for evaluating different methods of using LSA model for the task of document clustering. Below are some related works to our research:

The work of [10] compared the performance of LSA model with its probabilistic version known as Probabilistic Latent Semantic Analysis, PLSA. The task was to improve the clustering of documents in the Polish language. Although the authors did observe that the LSA performance was better than the PLSA model, by using the purity measure of cluster evaluation, which is one of evaluation methods used in our study as well. However, they did not pay attention to the parameters inside the LSA model itself for finding the best version of the LSA model for the same task of Polish text document clustering

The authors in [11] performed a detailed survey with 17 papers, on the different methodologies of semantic-based clustering of text documents. Meanwhile the authors

presented a comprehensive comparison of different semantic based document clustering approaches, but they just considered the LSA model with cosine similarity metric, which is one of the semantic similarity metrics being investigated along with others with the LSA model for the task of web document clustering.

Doan [12] performed an investigation with the LSA model with the goal of improving the latent semantic analysis performance. More precisely how the noise data affects the final results of the LSA model. While this work showed very clearly how noise data affects the performance of LSA model, such as the shared terms decrease the performance of LSA for matching queries to documents from the same category and they found that identification and elimination of shared terms is key to increasing LSA performance, but they did not focus on investigating which semantic similarity measure works better with the LSA model.

The authors in [13] used latent semantic analysis model for semantic based text document clustering. They proposed a genetic algorithm based on a latent semantic model (GAL), which is a technique based on natural selection. Two sets of data are experimented in this study; dataset 1 consists of 600 texts from three topics and dataset 2 contains 1000 texts from five topics, both datasets are taken from Reuter-21578 text collection. This work just used F-measure for evaluating the clustering algorithm results and ignored the purity and entropy metrics, and the similarity measure used in this study was only cosine similarity, as we have proposed five other similarity measures along with cosine similarity, for investigating the performance of LSA with all of them and finding the semantic similarity measure that works better with LSA model.

Zheng *et al.* [14] used latent semantic analysis and gene ontology (GO) for semantic based document clustering. The method used an ontological clustering method called GOClonto. The method used PubMed abstract collections as their data source and used GOClonto to conceptualize these PubMed abstract collections. As is mentioned in this study, the term conceptualization of PubMed abstracts means “representing PubMed abstracts with a set of key gene-related concepts and their relationships”. They also again did not use the purity and entropy metrics, and they just focused on F-measure for evaluation. Unlike, our proposed approach, this work is domain specific. It means their proposed work can only cluster text documents in medical datasets.

The authors in [15] also used the Latent Semantic Indexing LSI to cluster the programming source code files, which are technically in the form of text. Their applications for that source file clustering were: determining and identification of abstract data-types in procedural code files and the identification of concept clones. The other application was to determine traceability links between system documentation and the source code of the program. They used the LEDA (Library of Efficient Data Types and Algorithms) developed and distributed by Max Planck Institute für Informatik, Saarbrücken, Germany, which is a freely available C++ library, as their source of data.

The LSA approach is used by [16] as a base for learning topics from text corpus. Their input text corpus was manually tagged British National Corpus (BNC), which contains tens of millions of words taken as samples from both spoken and written English language. Then again, corpus words are taken from various sources of text such as newspapers, journals, and university essays as written part, and for spoken part. The BNC dataset is hand-labeled into nine domains for written text, the whole spoken text is collected under a single domain [16].

The LSA model is used for extracting corpus-level semantic space (lowering dimensional space), then it is used for the purpose of modeling different styling of text writing. The used text was like a transcription of informal conversation taken from volunteers from different social classes. As long as the authors used two semantic similarity metrics in their study; Euclidean Distance and Cosine similarity, but they followed a different approach and used them for a different purpose. They used British National Corpus in their study and made a comparison between hand-labeled domains and automatically generated classes. [16, 17].

The work in [18] used latent semantic analysis model to present a concept-based access to information. The presented approach was to adopt semantic relationships among concepts in corpus for the purpose of finding relevant documents. Also, the presented approach was used for the purpose of excluding irrelevant documents by recognizing semantic distance of concepts in corpus. Latent semantic analysis, on the other hand, attempts to reveal hidden conceptual relationships among words and phrases based on linguistic usage patterns. The main goal for this work was to explore the potential of concept-based semantics for accessing information, in other words for retrieving semantically related documents for a given document or text query.

The LSA model is used by [19] to present an approach for clustering text documents belonging to different languages. A parallel document corpus in both English and Chinese is collected from a thesis and dissertation digital library in Taiwan. The result of this Multi-Lingual Document Clustering (MLDC) was used for creating organizational knowledge maps.

The LSA model was able to produce organizational knowledge maps, which are document clusters for different topics existing in the corpus. The experimental results in this work showed that the proposed LSI-based MLDC technique achieved satisfactory clustering task. Therefore, the LSA model can be used for clustering text corpora that include multi-languages, such as text corpus of Wikipedia articles [19].

Hasanzadeh *et al.* [20] performed a different method for LSA-based clustering of text documents, which can be used in web search engines. The authors proposed a method of merging the LSA model with Particle Swarm Optimization algorithm (PSO), they named it as PSO+LSI. Their work was able to retrieve more relevant documents to the user query and avoid irrelevant documents, which belong to topics other than the topic of the given text query. For the purpose of clustering of documents, the authors used the PSO+k-means algorithm on the vector matrix produced from the LSA model.

After reviewing the literature due to the efficiency of the LSA model for semantic-based document clustering, our work presents a framework of investigating a proper usage of LSA model for the task of web document clustering. Our work investigates the effect of different semantic similarity measures with the LSA model, as the LSA model needs a semantic similarity measure to calculate the distance and similarity between the generated document vectors being generated by the LSA model.

2.2. Data clustering

Clustering is known as the most common unsupervised learning problem. It is a mechanism which subdivides a data set into number of clusters based on some given features in a feature space in a way that similar/related data objects are grouped together, whereas dissimilar/non-related objects are kept in separate clusters [21]. The Figure 2.1 shows a very simple example of the clustering process, in the Figure 2.1. (a), there is an

unlabeled data set and the data are separated as two groups, and in the Figure 2.1. (b), the data are partitioned into two coherent circle and square shaped clusters using an unsupervised learning algorithm, such as k-means.

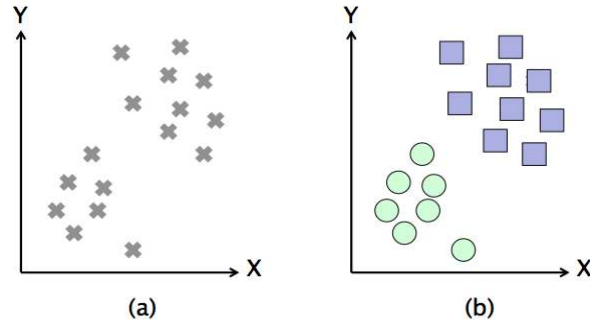


Figure 2.1. General view of data clustering [21]

It is significant to understand the distinction between supervised classification and unsupervised classification (clustering). In supervised classification, the data are labeled. On the other hand, for unsupervised classification, the problem is to partition a given set of unlabeled features into seriously meaningful partitions. Thus, the word “clustering” is a term used in data science communities to picture the methods for grouping of unlabeled data [21, 22].

The data clustering process involves five main components, Figure 2.2.

- a. Features Selection/Extraction: where interesting features are selected from the input data objects, as it is not the case, that all the features in the data are necessary for the clustering task.
- b. Pattern Representation: the input patterns might be transformed to a different representation rather than their main input representation, such as vectors or graphs.
- c. Measuring Similarity: one or more similarity (i.e. semantic similarity) measures are used in this step to measure the similarity and dissimilarity of the input data objects. This step is crucial for the further step or grouping and clustering of the input data objects.
- d. Grouping: this step includes the mechanism and algorithm selected for the real clustering of data. Data clustering algorithms come in different themes depending on the final needed results of the clustering task.

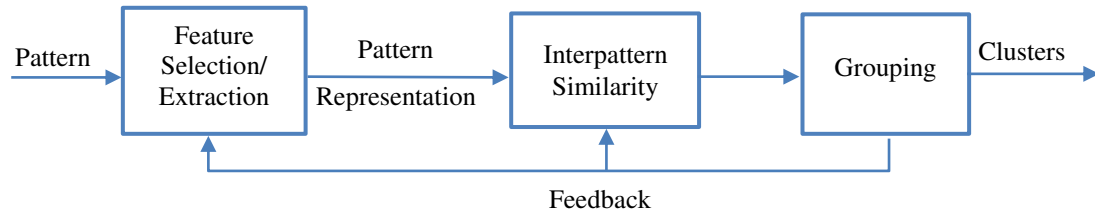


Figure 2.2. Main steps in data clustering [21]

Clustering has a long history among scientific societies. Clustering algorithms mainly categorized into two groups: partitional clustering algorithm and hierarchical clustering algorithm, Figure 2.3. A hierarchical algorithm splits the dataset into smaller portions (nested clusters) in a hierarchical form; partitional algorithm on the other hand, partitions the given dataset into a required number of clusters simultaneously [23].

The most commonly used and well-known partitional clustering algorithm is K-means. Many clustering algorithms have been published such the Single-link algorithm and complete-link algorithm which both are common hierarchical type clustering algorithms, and Fuzzy c-means algorithm, K-means algorithm, and Gaussian algorithm are samples of partitional clustering algorithms. Until now K-means is still from the most common and simplest clustering algorithms. Since publishing the k-means algorithm over sixteen years ago, many other clustering algorithms have been proposed, but still K-means is the popular clustering algorithm [23].

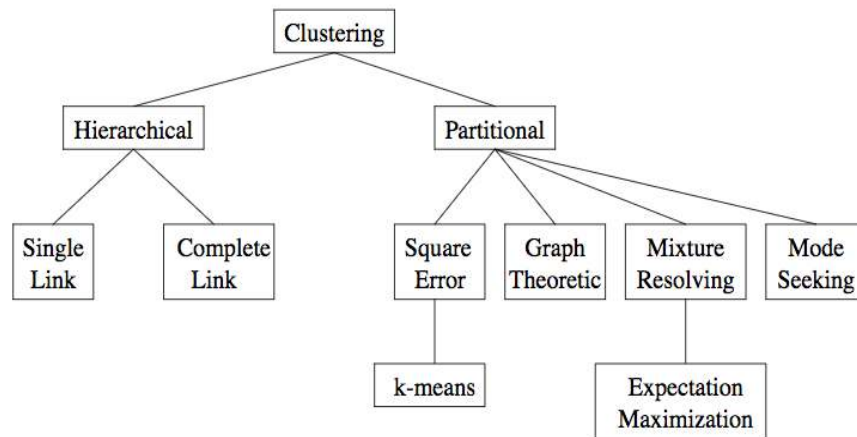


Figure 2.3. The taxonomy of data clustering approaches [21, 22]

2.2.1. Partitional Clustering Algorithms

Partitional clustering, is also sometimes called flat clustering. It differs from hierarchical clustering; hierarchical clustering performs one of two, either partitioning each cluster into smaller portions or agglomerating two similar clusters into a larger cluster. Whereas, in partitional clustering data collections are broken into independent partitions [24]. Partitional clustering divides the datasets into a desired randomly predefined K clusters and assigning a centroid for each cluster. The most popular partitional data clustering algorithm is k-means and its variants [24].

2.2.2. Hierarchical Clustering Algorithms

Hierarchical clustering algorithm tends to build a tree structure of clusters, which creates a series of nested sub-trees, and each leaf represents a cluster node of an object. There are two major methods used in hierarchical algorithms: agglomerative (bottom-up) and divisive (top-down) [24, 25].

2.2.2.1. Agglomerative Hierarchical Clustering (AHC)

In agglomerative clustering, each data object is defined to represent its own cluster. Each cluster has sub-clusters (nodes). Agglomerative clustering uses a bottom-up approach, it starts from a singleton cluster, during each step the two most similar/ nearest clusters are agglomerated iteratively into one cluster using some given measures creating a bigger cluster until a single cluster remains that contains all the merged document clusters [24, 26].

2.2.2.2. Divisive Hierarchical Clustering (DHC)

The divisive clustering works in the opposite of agglomerative clustering. It follows the top-down approach that all the documents are combined in the same cluster (all-inclusive), and then it recursively starts dividing each cluster into two smaller new child clusters until a singleton (a cluster with a single document) cluster level is reached or a certain criterion is met [24].

Data clustering could be further categorized into hard clustering and soft clustering [24]. In hard Clustering, each data object will be a member of only one of the generated

clusters, whereas soft clustering means that a data object may have a membership degree in multiple clusters.

2.3. Web Document Clustering

Web document clustering includes various methods for fast information retrieval purposes on the web [27]. The problem arises when the user has difficulty to find a desired topic among many non-relevant results of a searching query on web. Here the role of web document clustering becomes more obvious, which clusters web documents and categorizes them and makes them ready for user queries to retrieve the most relevant web documents to the user [25].

In the current web search engines, the grouping of web documents is mostly dependent on the overlapping of the keywords between them. Thus, web document clustering involves gathering the most similar web documents into one cluster and at the same time collecting different web documents into different groups or clusters, in which web documents of one cluster will always share some similar subjects differing from documents in another cluster [25].

2.4. Natural Language Processing

Since the input of the system is in natural language, it is necessary to pre-process it into a format, which helps in the steps afterward. Although CoreNLP provides many features to its user, but we are just using the features that our application needs, which are listed below:

- a. *Text Tokenization*: Tokenization is the initial step in the natural language processing, it works as tokenizing the inputted text documents into sequence of tokens, typically words, to make the input text simpler and easier to handle in the further advanced steps [1, 28].
- b. *Part of Speech Tagging*: To label the tokens with their part-of-speech (POS) tags, such as an identification of words as nouns, adverbs, verbs, and adjectives [28, 29].
- c. *Lemmatization*: lemmatization is about morphological analysis of the words, it generates the base forms (lemmas) for all tokens in the annotation [28, 29]. It

maps several words into one common root. For example, gone, going, and went into go.

- d. *Stop-word Removal*: they are words which are most frequently used in a context and they are less important than other words, such as in, on, at, and, the, and is. There are several lists of stop words containing different number of words. These words could be ignored or rejected after the POS tagging process as they are not nouns, adverbs, verbs, or adjectives [1].

The code segment below shows how the Stanford CoreNLP pipeline is used for parsing the input web documents. This piece of code simply reads the whole text from a text file, annotates its content, loops through the sentences, and through the tokens in each sentence, then finally prints the given token (word), its lemma (root word), and its POS tag [28]:

```
1: Properties props = new Properties();
2: props.put ("annotators", "tokenize, pos, lemma");
3: StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
4: File inputFile = new File("someInputText.txt");
5: String text = Files.toString(inputFile, Charset.forName("UTF-8"));
6: Annotation document = new Annotation(text);
7: pipeline.annotate(document);
8: List<CoreMap> sentences = document.get(SentencesAnnotation.class);
9: for (CoreMap sentence: sentences) {
10: for (CoreLabel token: sentence.get(TokensAnnotation.class)) {
11: String word = token.get(TextAnnotation.class);
12: String lemm= token.get(LemmaAnnotation.class)
13: String pos = token.get(PartOfSpeechAnnotation.class);
14: System.out.println("word: " + word + " , "+ "lemma: " + lemm + " , "+ " pos: " +
    pos); } }
```

The object *props* is the main Stanford CoreNLP object (Line 1), which enables the selection of needed *annotators* such as: "*tokenize, pos, lemma*" (Line 2). The selected annotators are assigned to the Stanford CoreNLP pipeline, using the object of *pipeline*

(Line 3). In Line 7 the input text from a text file gets annotated. The object *sentences* hold all the sentence from the text file (Line 8 and 9). The loop in Line 10, iterated all the words inside a given sentence, and prints its word (as appears in the main text), its lemma (root word), and its part of speech tagging Tag. To explain more let us assume having a text file including the text of: “*You are doing a good work in that restaurant.*” The code above produces the output below:

```
word: You, lemma: You, pos: PRP
word: are, lemma: be, pos: VBP
word: doing, lemma: do, pos: VBG
word: a, lemma: a, pos: DT
word: good, lemma: good, pos: JJ
word: work, lemma: work, pos: NN
word: in, lemma: in, pos: IN
word: that, lemma: that, pos: DT
word: restaurant, lemma: restaurant, pos: NN
```

2.5. Term-Document Text Representation

Term-document matrix is an unavoidable step toward the latent semantic analysis, explained in the next section. TDM matrix computes the terms occurrence frequency in each document in the dataset. In any document-term matrix, terms are represented as rows and columns corresponding to documents existing in the dataset. The documents are modeled as bag of words. Such as assuming a dataset consisting of nine documents as following, from [30]:

- D1: "A measure of the efficiency of a person, machine, factory, system, etc., in converting inputs into useful outputs"
- D2: "Productivity is computed by dividing average output per period by the total costs incurred or resources consumed in that period."
- D3: "Productivity is a critical determinant of cost efficiency"
- D4: "An economic measure of output per unit of input. Inputs include labor and capital, while output is typically measured in revenues and other GDP components."

D5: "Productivity is measured and tracked by many economists as a clue for predicting future levels of GDP growth."

D6: "Productivity gains are vital to the economy because they allow us to accomplish more with less."

D7: "Productivity is the ratio of output to inputs in production; it is an average measure of the efficiency of production."

D8: "The rate at which radiant energy is used by producers to form organic substances as food for consumers"

D9: "Productivity is commonly defined as a ratio between the output volume and the volume of inputs."

Then the term-document matrix for the above dataset is as in the following table:

Table 2.1. Term-document matrix for the assumed dataset

	D1	D2	D3	D4	D5	D6	D7	D8	D9
measure	1	0	0	2	1	0	1	0	0
effect	1	0	1	0	0	0	1	0	0
machin	1	0	0	0	0	0	0	0	0
factori	1	0	0	0	0	0	0	0	0
system	1	0	0	0	0	0	0	0	0
input	1	0	0	2	0	0	1	0	1
output	1	1	0	2	0	0	1	0	1
averag	0	1	0	0	0	0	1	0	0
cost	0	1	1	0	0	0	0	0	0
resourc	0	1	0	0	0	0	0	0	0
consum	0	1	0	0	0	0	0	1	0
econom	0	0	0	1	0	0	0	0	0
labor	0	0	0	1	0	0	0	0	0
revenu	0	0	0	1	0	0	0	0	0
gdp	0	0	0	1	1	0	0	0	0
predict	0	0	0	0	1	0	0	0	0
futur	0	0	0	0	1	0	0	0	0
growth	0	0	0	0	1	0	0	0	0
gain	0	0	0	0	0	1	0	0	0
accomplish	0	0	0	0	0	1	0	0	0
energi	0	0	0	0	0	0	0	1	0
produc	0	0	0	0	0	0	0	1	0
food	0	0	0	0	0	0	0	1	0

One of the main drawbacks with this representation with the term-document matrix is that it might be the case that some less frequent terms in the dataset have more importance than only considering their low frequencies. The term weighing called TF/IDF handles that

drawback [24], which is the most common weighted-term technique. The TF/IDF stands for Term Frequency/Inverse Document Frequency, which is the weight of each element in the matrix, it is a proportional representation of the number of occurrence of each term in every document, the limited occurrence times of the terms are up-weighted to reflect the importance of those terms.

In Text Mining, TF/IDF is used to take texts as an input and convert them into vectors, each vector contains the weight of each element from each document. TF/IDF replaces the term by document frequency in the TDM frequency matrix by its TFIDF (*Term Frequency Inverse Document Frequency*) weight, using the following formula [24]:

$$\text{TFIDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D) \quad (2.1)$$

where:

$\text{TF}(t, d) = \text{Term Frequency}(t, d)$: represents how many times the term t occurs in a given document d .

$\text{IDF}(t, D) = \text{Inverse Document Frequency}(t, D)$: this measures how much the term t is important in all the documents in the dataset (D) using the following formula:

$$\text{IDF}(t, D) = \log_2 \frac{N}{\text{DF}} \quad (2.2)$$

Such as: N represents the total number of documents, and DF is defined as the number of documents which contain the term t . Thus, $\text{IDF}(t, D)$ divides the total number of documents by the number of documents in which the term t appears, and then it takes the logarithm of the quotient. The table 2.2, illustrates the TFIDF weighted matrix for the above TDM frequency matrix, from [30]:

Table 2.2. TFIDF weighted matrix for the assumed dataset

Term	D1	D2	D3	D4	D5	D6	D7	D8	D9
measure	0.237917	0.000000	0.000000	0.475834	0.305211	0.000000	0.305211	0.0000	0.000000
effect	0.396528	0.000000	0.814116	0.000000	0.000000	0.000000	0.508685	0.0000	0.000000
machin	1.189584	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000
factori	1.189584	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000
system	1.189584	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000
input	0.237917	0.000000	0.000000	0.475834	0.000000	0.000000	0.305211	0.0000	0.488469
output	0.198264	0.254343	0.000000	0.396528	0.000000	0.000000	0.254343	0.0000	0.407058
averag	0.000000	0.763028	0.000000	0.000000	0.000000	0.000000	0.763028	0.0000	0.000000
cost	0.000000	0.763028	1.221174	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000
resourc	0.000000	1.526056	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000
consum	0.000000	0.763028	0.000000	0.000000	0.000000	0.000000	0.000000	0.8746	0.000000
econom	0.000000	0.000000	0.000000	1.189584	0.000000	0.000000	0.000000	0.0000	0.000000
labor	0.000000	0.000000	0.000000	1.189584	0.000000	0.000000	0.000000	0.0000	0.000000
revenu	0.000000	0.000000	0.000000	1.189584	0.000000	0.000000	0.000000	0.0000	0.000000
gdp	0.000000	0.000000	0.000000	0.594792	0.763028	0.000000	0.000000	0.0000	0.000000
predict	0.000000	0.000000	0.000000	0.000000	1.526056	0.000000	0.000000	0.0000	0.000000
futur	0.000000	0.000000	0.000000	0.000000	1.526056	0.000000	0.000000	0.0000	0.000000
growth	0.000000	0.000000	0.000000	0.000000	1.526056	0.000000	0.000000	0.0000	0.000000
gain	0.000000	0.000000	0.000000	0.000000	0.000000	2.442347	0.000000	0.0000	0.000000
accomplish	0.000000	0.000000	0.000000	0.000000	0.000000	2.442347	0.000000	0.0000	0.000000
energi	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.7492	0.000000
produc	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.7492	0.000000
food	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.7492	0.000000

2.6. LSA Model

The weighted term-document generated by the TFIDF technique is large based on the number of unique terms in the dataset, to reduce the semantic space in the TFIDF matrix LSA model is used. The LSA model reduces the size of TFIDF matrix and thus approximates it to one of its lower ranks using the Singular Value Decomposition (SVD) technique in algebra [31]. The low-rank approximation to TFIDF matrix gives a new representation to each document in the corpus [24], with the latent semantics and topics from the whole dataset.

Then the approximated and reduced matrix is used to compute the semantic similarity between documents in the dataset. This process is sometimes also called as Latent Semantic Indexing LSI [24]. In linear algebra, the singular value decomposition transforms a given matrix into a bi-diagonal $M = U\Sigma V^T$ form. Figure 2.4 explains the process of LSA model.

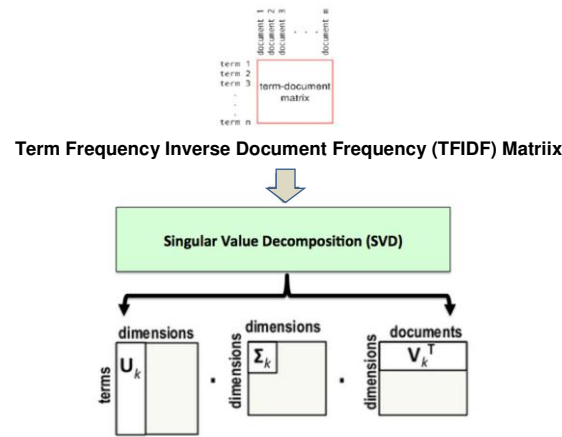


Figure 2.4. LSA model

To understand how the LSA works, let us consider the following working example from [24]. For simplicity consider the following term-document frequency matrix C which is an $m \times n$ matrix:

Table 2.3. Term-document matrix

C	D1	D2	D3	D4	D5	D6
Ship	1	0	1	0	0	0
Boat	0	1	0	0	0	0
Ocean	1	1	0	0	0	0
Wood	1	0	0	1	1	0
Tree	0	0	0	1	0	1

After running the singular value decomposition technique, its value would be the product of three matrices (U , Σ , and V^T) as below:

First matrix is U , which is called the SVD *term* matrix or the left singular matrix (an $m \times m$ square matrix):

Table 2.4. SVD term matrix

U	1	2	3	4	5
Ship	-0.44	-0.30	0.57	0.58	0.25
Boat	-0.13	-0.33	0.59	0.00	0.73
Ocean	-0.48	-0.51	0.37	0.00	-0.61
Wood	-0.70	0.35	0.15	-0.58	0.16
Tree	-0.26	0.65	-0.41	0.58	-0.09

Then the singular value matrix (the diagonal matrix) is:

Table 2.5. Singular value matrix

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

Eventually, there is V^T , it is defined as the right Singular Value Decomposition (SVD) document matrix in the context of a term-document matrix:

Table 2.6. SVD document matrix

V^T	D1	D2	D3	D4	D5	D6
1	-0.75	-0.28	-0.20	-0.45	-0.33	-0.12
2	-0.29	-0.53	-0.19	0.63	0.22	-0.41
3	0.28	-0.75	0.45	-0.20	0.12	-0.33
4	0.00	0.00	0.58	0.00	-0.58	0.58
5	-0.53	0.29	0.63	0.19	0.41	-0.22

Now, let us perform the real step of reducing the size of the original term-document frequency matrix by 2 (reducing the dimensions from five to two dimensions). First, we should zero-out all singular values and just keep the two largest values of the matrix Σ , so we obtain Σ_2 :

Table 2.7. Reduced singular value matrix

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

Then to get the final reduced C_2 , we perform the multiplication among the reduced Σ_2 , U and V^T :

$$C_2 = U\Sigma_2V^T \quad (2.3)$$

This way the final reduced term-document matrix is as the follows:

Table 2.8. Reduced term-document matrix

C₂	D1	D2	D3	D4	D5	D6
1	0.85	0.52	0.28	0.13	0.21	-0.08
2	0.36	1.36	0.16	-0.20	-0.18	-0.18
3	1.01	0.72	0.36	-0.04	0.16	-0.21
4	0.97	0.12	0.20	1.03	0.62	0.41
5	0.12	-0.39	-0.08	0.90	0.41	0.49

Now, we could view that C_2 is represented as a two-dimensional matrix, after performing a dimensionality-reduction on matrix C from five dimensions into two-dimensions, using the singular value matrix of Σ_2 , which is produced by the SVD technique.

Now let us observe the importance of the LSA model in using the SVD technique by the example below, which computes the semantic-similarity of two documents of d_2 and d_3 :

- The similarity between d_2 and d_3 in the original space, in the Matrix C : 0.0!
- However, the similarity between d_2 and d_3 in the reduced space, in the matrix C_2 : $\approx 0.52!$

$$= 0.52 * 0.28 + 1.36 * 0.16 + 0.72 * 0.36 + 0.12 * 0.20 + ((- 0.39) * (- 0.08)) \approx 0.7$$

Now our investigated research problem is: “*how to compute the semantic similarity to find the most accurate semantic similarity and relatedness between any two given web documents in the dataset for the task of semantic-based web document clustering?*”

For that purpose, we used the LSA model with a set of well-known semantic-similarity measurements to find which semantic-similarity measure works best with the LSA model for the task of web document clustering.

2.7. Text Semantic Similarity Measurements

This subsection describes different text semantic similarity measures [32-36], which have been used in the literature for performing different semantic-based text mining tasks.

- *Euclidean Distance*: Euclidean distance is a standard distance measurement used for geometrical problems. It calculates the distance between any two points. Euclidean distance is extensively used in clustering problems, and it is the default distance measure in the k-means algorithm.

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (2.4)$$

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \quad (2.5)$$

Where:

$p = \{t_1, t_2, t_3, \dots, t_n\}$ and $q = \{t_1, t_2, t_3, \dots, t_n\}$: are two points, representing two term vectors, to find the distance between them in the space.

Although being the default similarity measurement for the k-means algorithm, it is our task to discover how it performs for text document clustering compared to other semantic similarity metrics.

- *Jaccard Coefficient*: This measure which is also interchangeably named as *Jaccard Index*, considers the two document vectors as two sets of terms, then it uses their commonality and dissimilarity measures, see Figure 2.5, as shown in the following formula:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (2.6)$$

(If A and B are both empty, we define $J(A, B) = 1$.)

$$0 \leq J(A, B) \leq 1.$$

Where:

$J(A, B)$ value falls between 0 and 1. It would be 1 when $A = B$, and 0 when they are dissimilar (completely different), disjoint sets of terms.

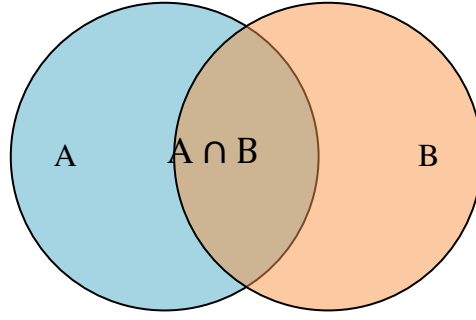


Figure 2.5. Intersection and union of two finite sets

- *Cosine Similarity:* Being documents defined as term vectors, the similarity between two documents can be corresponded to the correlation between those two vectors in the semantic space generated by the LSA model, which is the cosine value of the angle generated by the vectors, Figure 2.6. The cosine similarity is computed using the following formula:

$$sim(A, B) = \cos(\theta) = \frac{A \cdot B}{||A|| ||B||} \stackrel{\text{def}}{=} \frac{\sum_{i=1}^n (A_i * B_i)}{\sqrt{\sum_{i=1}^n A_i^2} * \sqrt{\sum_{i=1}^n B_i^2}} \quad (2.7)$$

Where: A and B : are two n -dimensional term vectors representing two documents of the term set of $T = \{t_1, ..., t_n\}$. The similarity values generated by the cosine similarity are non-negative values and in the range of 0 to 1, meaning from (0) meaning totally dissimilar documents (term vectors) to (1) meaning totally identical documents (term vectors).

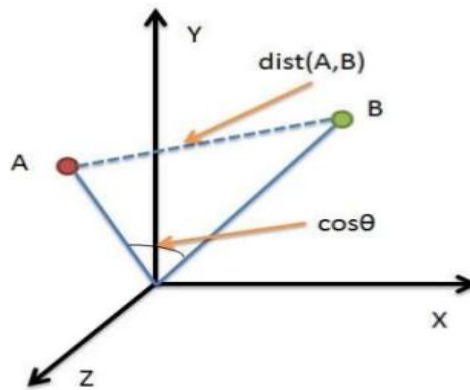


Figure 2.6. The concept of cosine similarity

- *Pearson Correlation Coefficient (PCC)*: Pearson's correlation coefficient is also another measure, which is used to calculate to how much degree two vectors are related, document vectors in our case. It is commonly used in statistics to find the correlation between sets of data, by measuring how well the data are related to each other.

More precisely, it measures the linear correlation between two variables, such as X and Y . The Pearson correlation coefficient formula has different formalizations.

$$PCC(X, Y) = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n(\sum x^2) - (\sum x)^2][n(\sum y^2) - (\sum y)^2]}} \quad (2.8)$$

Where:

n is the number of dimensions the documents have been represented (i.e. 300)

X and Y are two document term vectors of the length n -dimensions, such as:

$$X = \{x_1, x_2, x_3, \dots, x_n\} \text{ and } Y = \{y_1, y_2, y_3, \dots, y_n\}.$$

$\sum x$: sum of x (weights of) terms

$\sum y$: sum of y (weights of) terms

$\sum xy$: sum of products of (weights of) paired terms

PCC has the values in the range between +1 to -1, where (+1) means that the linear correlation is completely positive, (0) means there is no linear correlation, and (-1) means that there is the linear correlation which is completely negative. Examples of scatter diagrams with different correlation coefficient (ρ) values are visually presented in Figure 2.7.

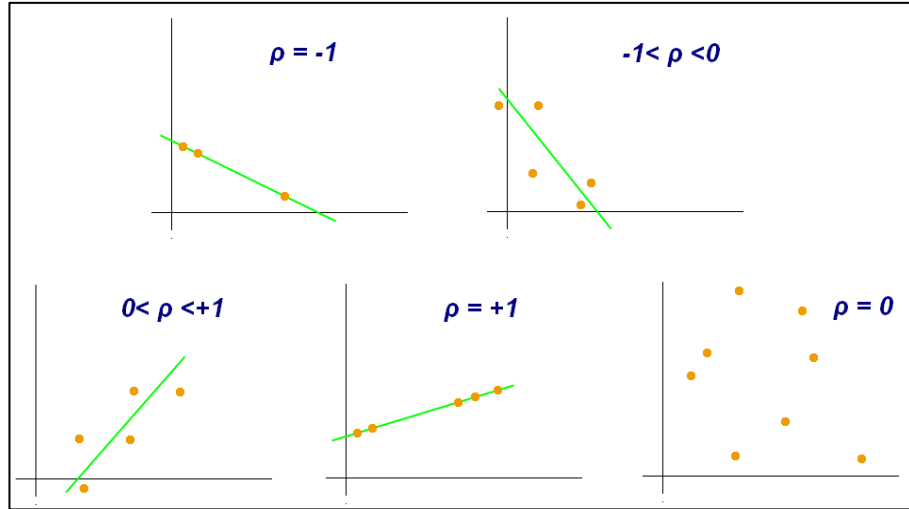


Figure 2.7. Scatter diagrams for different correlation coefficient values

- *Dice Coefficient*: is like the Jaccard coefficient, the Dice coefficient similarity also considers the two document vectors as two sets of terms, as twice the number of common terms in the compared documents divided by the total number of terms in both documents and computes the similarity of two document vectors using the following formula:

$$\text{Dice}(A, B) = \frac{2|A \cap B|}{|A| + |B|} \quad (2.9)$$

Where:

$|A|$ and $|B|$ are the numbers of elements in the two documents (term vectors / data samples) of A and B .

$|A \cap B|$ is the size of the intersection sub-set of two given documents.

- *Feature_Overlap Coefficient*: is another similarity measurement technique that is relevant to the Jaccard index. It is used to measure the overlap between two sets of terms, and is defined as the size of the intersection of the two vectors divided by the smaller of the size of the two vectors:

$$\text{Feature_Overlap}(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)} \quad (2.10)$$

Where *Feature_Overlap* (A, B), is the size of the intersection sub-set of two given documents divided by the size of the document with minimum size (number terms).

2.8. K-Means Algorithm

Data clustering in general aims at partitioning n data points into k -clusters. Thus, it aims to assign a cluster to each data point. K-means is a famous clustering algorithm that aims to find the positions n *centroids*; the data points, which represent the centers of the n cluster, in a way that minimizes the distance from the data points to the cluster centroids.

The very first step of k-means is to begin with randomly initializing a fixed number of clusters, say k , and define centroids for each cluster. Then moving these centroids through finite number iterations until it converges (no changes happen to the locations of centroids) [24]. The algorithm below explains in detail the steps involved in the k-means algorithm [37].

As explained in the algorithm below, and generally in the data clustering problem, we are given an input dataset of m elements: $x^{(1)}, \dots, x^{(m)}$, we aimed to partition and cluster the input data points into k number of groups. Here, we are given feature vectors for each data point $x^{(i)} \in \mathbb{R}^n$ as usual; but no labels $y^{(i)}$ (it makes this an unsupervised machine learning problem). We aim to predict k centroids and a label $c^{(i)}$ for each data-point [37].

1. Initialize **cluster centroids** $\mu_1, \mu_2, \dots, \mu_k, \in \mathbb{R}^n$ randomly.

2. Repeat until convergence: {

For every i , set

$$c^{(i)} := \arg \min_j ||x^{(i)} - \mu_j||^2 \quad (2.11)$$

For each j , set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)}=j\}x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)}=j\}} \quad (2.12)$$

}

Ultimately, k-means will find the typical centroids (locations) by cycling between (1) assigning data objects to clusters based on their *distance* to current centroids. (2) for each cluster, choosing new centroids (data-points which are the center of a cluster) based on the current assignment of data points to clusters [37]. The figure 2.8 visually explains the k-means algorithm through step by step iterations [37].

In the given data-clustering example, in Figure 2.8, the work of k-means algorithm is simply explained. Input data-points are shown as green spots in (a), and centroids of the clusters are shown as red and blue x (b). Where: (a) Actual dataset. (b) Randomly initialize cluster centroids. (c-f) illustrate the running of four iterations of k-means. At every stage, we assign each input data-point into the nearest centroid of the cluster (as shown by coloring the data points with the same color as the cluster centroids are assigned to); then we find a new centroid (cluster center data-point) for each cluster, by moving cluster centroids to the mean of the points assigned to it.

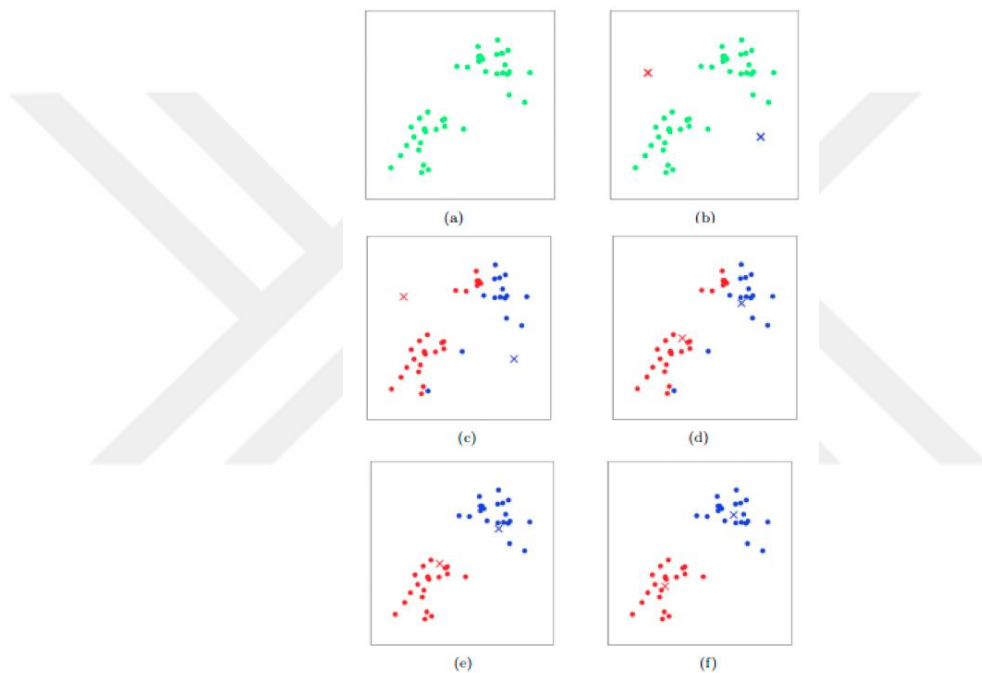


Figure 2.8. K-means example [37]

Our only modification inside k-means algorithm is to change the distance/similarity measure every time in the step two, in the above algorithm, to test their performance in clustering document vectors after being reduced by the LSA model. This way we can investigate and find the semantic-similarity/distance measure which best work for the LSA model when it is used for the task of data clustering.

The K-means algorithm requires three main user specified arguments [23]: number of clusters k , cluster initialization, and distance metric. Different initializations of K clusters could lead to different final clustering results. The most common way is to randomly

choose the number of clusters k , unless we have some idea from our application domain about the nature of the clusters we are going to produce by the k-means algorithm [23].

As far as for the distance and similarity measure, typically k-means algorithm integrates the Euclidean metric to measure the distance between cluster centers and points, however this is our task to test the algorithm performance with different similarity and distance metrics.



3. TEXT MINING

In general, the term “text mining” is used to refer to any system that is used to analyze in most of the cases large volumes of natural language texts, also observing their linguistic structure and endeavoring to extract useful information from those texts [2, 38], such as topics. Text mining is one of the main fields in computer science, which handles the problems encountered in industry when having sources of natural language and mainly textual data as their core input, such as Twitter and Wikipedia. Text mining also referred to as text-oriented data mining, which aims at extracting knowledge from textual data, for looking for patterns and trends inside textual data for a given application area, such as business and politics [39, 40].

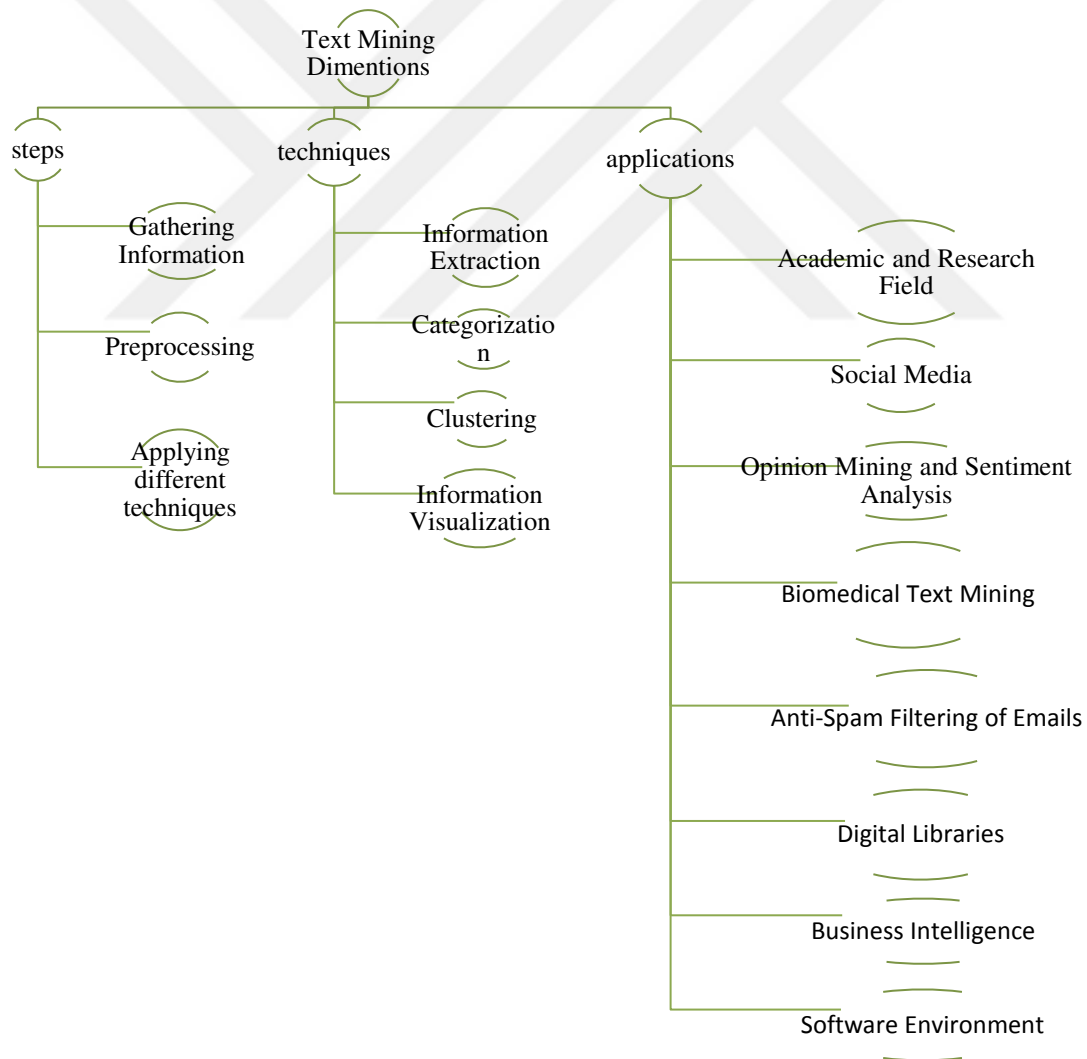


Figure 3.1. Text mining dimensions

As explained earlier, it is clear that the textual data is the most common form of stored information. It is believed that text mining has a higher commercial potential compared to other branches of data mining. Indeed, according to a recent study it is indicated that text documents occupy 80% of a company's information, thus requiring more efforts to extract more insights from it [41, 42]. The above figure is the illustration of the three text mining dimensions we have mentioned in the following sections, Figure 3.1.

3.1. Steps in Text Mining

In general, we could say that text mining is a branch of data mining, but when we talk about data mining specifically, it is used to extract knowledge from structured data such as relational databases (rows and columns) [39, 42]. Whereas, text mining also works to extract valuable information from unstructured texts (typically blog posts, emails, text documents, articles, social media messages, etc.) or semi-structured data sets such as HTML and XML files, etc. [43].

Figure 3.2 shows the general steps needed for performing text mining tasks. It starts with collecting unstructured textual data from the source of information, such as bloggers, social networks, or websites. Then it pre-processes it to transform it to a structured form, such as vectors, thus to be more efficiently processable by text mining algorithms. Finally, based on the application domain, one or more text mining algorithms are selected to perform the required task for the given final goal of the application, such as text clustering, summarization, and classification.

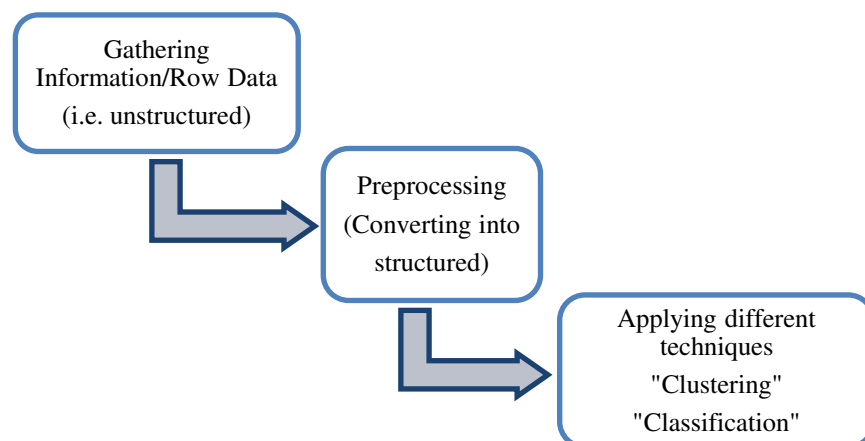


Figure 3.2. Basic process of text mining

3.2. Principal Fundamental Text Mining Techniques

As there are numerous sources that generate textual data, to get the best benefit from those textual data, many text mining techniques are offered for analyzing given text documents and extracting valuable information for different purposes; some important and widely used techniques are discussed below.

3.2.1. Information Extraction

Information Extraction is a technique of extracting valuable and most relevant textual units, e.g. phrases and the relationships in a large volume text documents and text corpora. Text corpora are text resources, which might be structured or semi-structured or even unstructured [44]. Information Extraction is commonly used for discovering structured information inside unstructured or semi-structured text documents [45] such as taxonomical relations among terms in corpus.

The extracted documents are saved into databases for subsequences processes needed [46], called text corpus. In another way, we can consider the information extraction as a fixed form of natural language understanding, while we already know the information we are looking for [47]. For instance, the following sentence:

“Amazon was found by Jeff Bezos in July 5, 1994”

The following information could be extracted from the given sentence:

Founder-Of (Jeff Bezos, Amazon)

Founded-In (Amazon, July 5, 1994)

3.2.2. Categorization

While the online information is in a rapid and constant growth, the text categorization [48] aims to classify and organize the documents into a finite set of predefined, may be structured, categories [44]. For example, the techniques of text categorization are utilized for classifying news articles and stories, to extract interesting and meaningful information and insights from the World Wide Web, and to guide the user’s searching queries over the web through hypertexts.

Since constructing text classifiers manually by hand is not easy and is time consuming, it is an advantage to use text classification algorithms to classify text corpora in a notably short time and with precision [48].

3.2.3. Clustering

Clustering is an unsupervised technique used for grouping the text documents into meaningful groups, in which similar documents representing a specific topic are grouped together and different documents belonging to different topics are kept in separate groups [44, 46]. Text clustering is differed from categorization in that the topics are not predefined to the system. The documents are represented as a series of vectors of tokens or words [2]. Different clustering algorithms are used in terms of text resources, such as K-means algorithm, Agglomerative algorithm, Fuzzy K-means.

3.2.4. Information Visualization

Information visualization, or sometimes called visual text mining, is the process of constructing visually explained structures such as a hierarchy or map from inputted big textual sources to provide browsing efficiency, as well as simplifying searching in the web [44]. The information visualization system visually gives insights and interesting information to its end user. Most of the times information visualization systems come with navigations tools, which make the visual exploration of the visualized information much easier to the end user.

3.3. Text Mining Applications

There are numerous datamining applications that have been invented which are used in many scientific communities and industry domains [49]. In this section we discuss some of the real life and commonly used applications of some text mining techniques.

3.3.1. Academic and Research Field

In the field of education, there are many text mining techniques and tools that are developed for the purpose of analyzing educational and academic materials and resources. As widely used text mining techniques, k-means and other related text-clustering algorithms have showed a good rule in the research field. It benefits students and

researchers finding relevant research papers, articles, and other academic materials [46], such as IEEE and Springer.

3.3.2. Social Media

Recently the volume of text data in social media has seen an unprecedented growth by increasing the number of social media platforms and the number of individuals using these social media. Many text-mining tools are proposed and available in terms of social media analysis. Social media analysis involves the textual outputs such as; emails, social networks, blogs, etc. For example, in case of social media networks, such as Facebook, there are well designed algorithms and tools used for observing the number of likes and comments, sharing of a post, and the number of followers, this helps researchers and industries to see the interaction and reactions toward different trends and subjects over the posts [46]. Also, text-mining algorithms have helped in understanding user social behaviors when they interact in the cyber-society, using novel text analysis approaches.

3.3.3. Opinion Mining and Sentiment Analysis

As the name indicates, opinion mining and sentiment analysis applications look for discovering the opinion and mood of the users, such as customers, on a specific product, such as web services or car products. These algorithms analyze the textual data produced by the users as their review on products on the web, such as in Amazon [47].

Such text mining applications enables the businesses to grow fast by knowing what products are mostly liked by their customers and which are not. For example, the companies can find remarkable information and opinions about a given topic, which is principally substantial in advertising and online marketing [47].

3.3.4. Biomedical Text Mining

Perhaps, biomedical text mining is one of the most active applications of text mining. It takes as input text of biomedical sciences domains, such as biomedical academic articles and patient reports and records. Biomedical text mining enables the biomedical researchers to efficiently manipulate knowledge from large volumes of data, mostly textual data, also facilitates biomedical discovery by analyzing genome sequences and protein structures for

different purposes such as; drug-to-drug interactions [47]. This helps in proposing new and more efficient treatments for already known diseases.

3.3.5. Anti-Spam Filtering of Emails

Security and privacy are other important applications of text mining, which mainly works on e-mail services, which are more commonly known as web services. E-mail security services attempt to capture and recognize harmful emails, known as spam emails. One common solution is the anti-spam filters. Whereas, the most commercially available filters such as blacklists and human-made rules of filtering [49].

Text classification in machine learning has notably helped when offering efficient anti-spam filters that may quickly adapt to new kinds of spam, after being trained. Spam filtering systems have been modeled in different ways, and mostly using statistical naive Bayes models. Perhaps, Mozilla's e-mail client is a major example [49].

Michelakis *et al.* [50] compared various classifier methods and investigated various costs of classifying a proper email as spam. The authors found that for their benchmark corpora, the support vector machine (SVM) almost always produces the best outcomes. They concluded that “these good results may be improved by careful preprocessing and the extension of filtering to different languages” [49].

3.3.6. Digital Libraries

In digital libraries, such as journals and proceedings of conferences they have valuable content, which are all in the form of textual digital data. Therefore, numerous text mining tools and techniques have been proposed and developed to extract and learn interesting insights from them. Digital libraries as sources of human knowledge have benefits in the domain of research and development, as they are great and significant sources for getting interesting information for the researchers [46].

Digital libraries organize information in such a way that makes it possible to efficiently make trillions of documents available online. Green-stone is one of the international digital libraries that supports accessing millions of text documents available online, it also has the flexibility to support multiple languages and multilingual interfaces

that can provide a solid method to extract multiple format documents, i.e., e-mail messages, Microsoft word, PDF, and HTML, it also supports the document extraction in the form of audiovisual and image format along with text documents. However, when text mining comes to the stage, digital libraries become much more interesting [43, 46, 51].

As text-mining algorithms make digital datasets to become organized more efficiently and the information retrieved from them is more accurate and faster. Mostly, in the text mining process various text manipulations occur such as: documents selection, enrichment, extracting information and tackling entities among the documents and generating instinctive co-referencing and summarization. In practice, GATE, Net Owl and Aylien are frequently used tools for text mining in digital libraries [46, 51].

3.3.7. Business Intelligence

Organizations and enterprises also generate numerous amounts of textual data for their business correspondence and transactions. All these textual data hold valuable insights for their owners. To extract this valuable information from their textual data, business owners need intelligent text mining algorithms, which can play a significant role in business intelligence by analyzing their customers and competitor's interactions for making better decisions [46].

Text mining tools provide a deeper insight about business and give insights on how further to improve their products. The text mining tools like IBM text analytics, Rapid miner, GATE help to take smart decisions about the organization itself and their services and products. These systems generate alerts about good and bad performance, market changeover that help to take remedial actions. Furthermore, it helps in the telecommunications industry, business and commerce applications and customer chain management system [46, 52].

3.3.8. Software Environment

Since the main data format in software also is text, therefore, smart text mining techniques are also being studied, proposed, and developed by larger companies such as IBM and Microsoft, to further automate the mining and review of their software development processes. They use text-mining algorithms to explore and index their

projects to improve their results. Within the public sector much effort has been concentrated on creating software for tracking and monitoring terrorist activities [39].

3.4. Semantic-Based Text Mining

As hidden patterns must be discovered through text mining algorithms, it is necessary for those algorithms to understand the meanings, in other words semantics of words in the text document or text corpus. The reason is to have more intelligent text mining algorithms that enable more intelligent text analysis applications.

When it comes to textual data, the anticipated systems are expected to be able to learn and represent semantics (meanings) of the terms of the text in a proper way, which enables better performance of the given text-based application, such as text summarization, classification, and clustering. For this purpose, the statistical models have been proposed such as latent semantic analysis model (LSA), which uses feature vectors for representing the semantic space.

The LSA model has become the leading semantic model for text mining and more specifically for learning and representing the semantics from the text corpus [53]. The performance of LSA model, as a model of semantic learning, has been ambiguous after being used with different available semantic similarity metrics for web document clustering. Therefore, there is a need to investigate the different variations of using LSA for the task of web document clustering. The importance of this study is to find the best combination of LSA model with semantic similarity measurements.

3.5. LSA-Based Text Document Clustering

Using LSA model for text clustering follows a general pipeline of main phases. It starts with natural language pre-processing, vector-based transformation, SVD decomposition, semantic similarity metric selection and k-means algorithm. This section describes these main phases as follows.

- *Natural language pre-processing of the textual document dataset:* this pipeline is necessary for almost any text analysis and mining task. Because the text needs to be transformed to a better format, which is more suitable

for the afterward steps of text mining, such as text summarization and clustering.

- *Constructing the tf/idf matrix:* this step is also necessary for finding the initial semantic similarity among documents in the corpus. In this step documents are converted to vectors and a unique list of terms for the whole corpus is generated.
- *Construction of the semantic space using the LSA model:* the TF/IDF matrix needed to be reduced by size and normalized so the values inside it better represents the semantic similarity of documents.
- *Semantic similarity metric selection:* this step determines a semantic similarity metric which is used to calculate the distance or similarity between documents in the semantic space generated after using the LSA model.
- *Using the k-means data clustering algorithm:* K-means is the main and most popular data clustering algorithm. In our work, for our study we are using different semantic-similarity metrics, to perform the document clustering task.

Chapter 5 describes the usage of these phases for investigating the most efficient semantic similarity metrics with LSA model for the task of semantic-similarity web document clustering.

4. PROBLEM STATEMENT

This section states explicitly the problem for which we are intending to propose a solution for, which is the semantic-based web document clustering. In the semantic-based web document clustering, the unlabeled web documents are grouped into semantically related clusters, where the web documents in the same cluster belong to the same or similar topics, whereas the web documents in different clusters belong to different topics. The input web document dataset is unlabeled, which means the web documents have no explicit indications to what topic each web document belongs. Therefore, the process of semantic-based web document clustering became an unsupervised machine learning task.

4.1. Semantic-Based Web Document Clustering

As in all data clustering tasks, in semantic-based web document clustering as well, some basic elements should be available in the approach. First, the semantic modeling approach that extracts the latent meaning from the input corpus. Second, there should be a measure of similarity or dissimilarity involved to find the degree of the relatedness of any given pair of web documents in the dataset. The semantic modeling approach being investigated in our work is the LSA model, as the semantic learning and knowledge representation model [31]. It is based on a general mathematical learning approach that accomplishes capable inductive impacts by extracting the proper number of dimensions that represents the topics inside the dataset [31].

4.2. LSA-Based Web Document Clustering

The LSA has been applied successfully in many text-mining and information retrieval tasks. When used for the text-clustering in general and web document clustering task specifically, the LSA model performance depends on the semantic similarity measurements being used with it. Therefore, there is a set of possible variations for using the LSA model for semantically clustering the web documents, where different possible semantic similarity measures can be used for the afterward step of clustering document vectors by the k-means algorithm.

So, it is our research problem here to investigate and find the best possible variation of using the LSA model with semantic similarity measures for semantic-based web document clustering task. To do so, we have used the k-means data-clustering algorithm, as the most famous data-clustering algorithm in the literature.



5. PROPOSED APPROACH AND METHODOLOGY

The approach we are proposing is to test different possible combinations of using semantic-similarity measurements with the LSA model for the web document-clustering task. This section demonstrates the proposed approach, the performance evaluation methods for the implemented system, and the experimental settings being used in our approach.

5.1. Proposed Approach Steps

The steps in our approach and its dataflow are illustrated in Figure 5.1. The steps in our method are as follows:

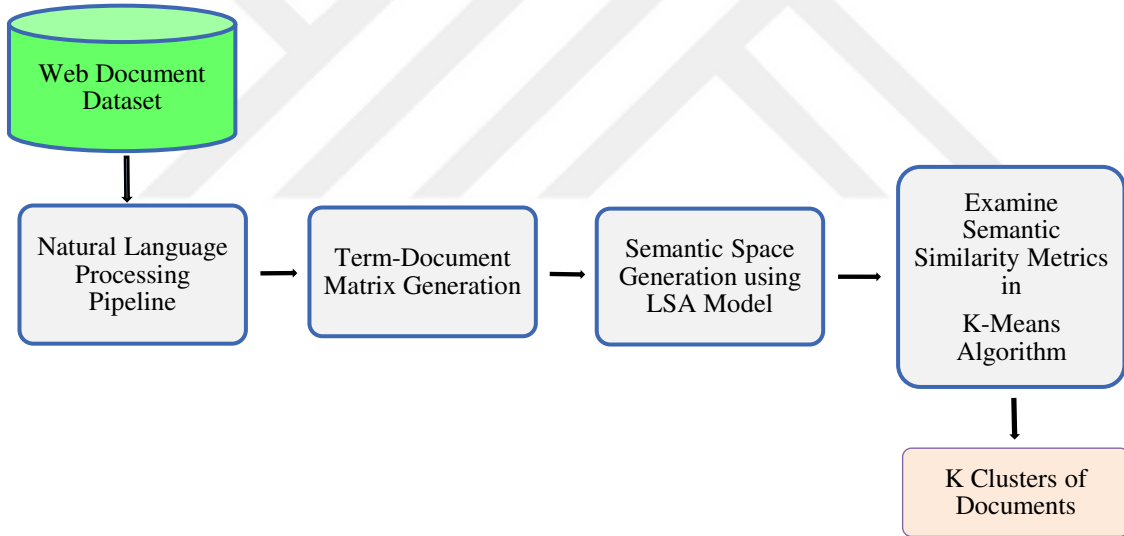


Figure 5.1. Steps of the proposed approach

5.1.1. Natural Language Processing (NLP) Pipeline

To pre-process the input text documents our system used the Stanford CoreNLP pipeline [28]. Generally, NLP aims to use technology (using computers) to improve the understanding of natural language better [54]. For performing the main NLP steps described in the background section, chapter 2.

5.1.2. Term-Document Matrix (TDM Matrix)

After preprocessing text documents, they are ready for being transformed from raw text documents into feature-based vector representation. We have used the TF/IDF term weighting scheme described in the background section, in which terms are represented as rows and columns correspond to documents existing in the dataset. The documents are modeled as bag of words.

5.1.3. LSA Model

LSA model reduces the size of TFIDF matrix and thus approximates it to one of its lower ranks using the Singular Value Decomposition (SVD) technique in algebra [14]. The mathematical foundation for LSA model is described in the background section, chapter 2.

5.1.4. Investigated Semantic Similarity Measurements

Semantic similarity measurements calculate and finds the degree of similarity/relatedness between a given two vectors, in our case documents vectors of terms. There have been a notable number of distance and similarity measures used for the task of document clustering, such as Jaccard Coefficient, Squared Euclidean Distance, and Cosine similarity. The LSA model has already been used for the task of document clustering, where the TF/IDF matrix is reduced and normalized to a new matrix (new set of document vectors).

However, it has been an open question of: *which semantic-similarity measures works best with the LSA model for the task of (web) document clustering?* This is our research question, which we are trying to handle by using real-world datasets of web documents. A list of investigated semantic similarity measurements are described in detail in the background section, chapter 2.

5.1.5. K-Means Algorithm

We are using *k-means* algorithm in our work to cluster the web documents on semantic bases. Each document vector from the reduced *term-document matrix* from LSA model is used as a data-point in the semantic space.

5.2. Cluster Evaluation Methodology

By using data and document clustering evaluation methods we evaluate the clusters produced for each variations of the combinations of LSA model and the six-investigated semantic-similarity metrics. The general evaluation methods in data clustering determine the aim of obtaining high intra-cluster similarity (the documents inside a category are similar) and low inter-cluster similarity (the documents within different categories are dissimilar) [24]. This is what is called an internal criterion for the quality of a clustering. We also used the entropy data clustering measurement, which evaluates the overall quality of the document clustering process, by investigating the normal distribution and accuracy of documents over produced clusters.

A gold standard dataset is used, which is ideally produced by human judges with a good level of inter-judge agreement [24]. Then evaluate how well the clustering matches the quality of the gold standard classes. To measure the quality and accuracy of our clustering results, two popular and widely used evaluation methods are used; purity and entropy. The *purity* metric is a transparent and simple data clustering evaluation measure of type of *external* [24]. To perform the purity, the clusters are assigned to the classes which are most common (appeared) in each cluster, after that to check for veracity of the performed assignment we will count the number of properly assigned classes (documents) and divide it by N .

An example on how the purity computes in Figure 5.2 [24], when the dominant class and number of members of the dominant class for all three clusters are denoted by symbols, such as five stars (*) (in the cluster 1); four of (%) symbol (in the cluster 2); and 3 of (^) symbol (in the cluster 3). Therefore, *purity* for this given data clustering result is $(1/17) * (5 + 4 + 3) \approx 0.71$.

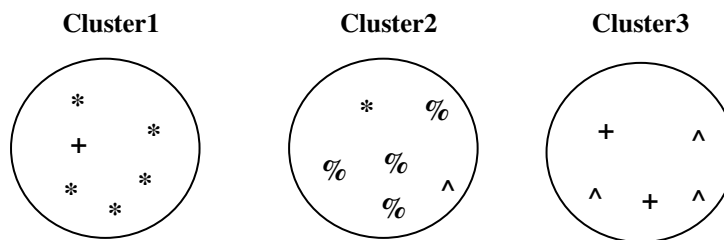


Figure 5.2. Purity as an external evaluation criterion for cluster quality [24]

The *purity* evaluation measure, of type of external criterion, is formally defined in the following formula [24]:

$$purity(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j| \quad (5.1)$$

Where:

- $W = \{\omega_1, \omega_2, \dots, \omega_K\}$ is the set of clusters.
- $C = \{c_1, c_2, \dots, c_J\}$ is the set of classes. ω_k is interpreted as the set of documents in ω_k and c_j is also interpreted as the set of documents in c_j in equation (4.1) [24].

$$Entropy(C_i) = -\frac{1}{\log c} \sum_{h=1}^k \frac{n_i^h}{n_i} \log \frac{n_i^h}{n_i} \quad (5.2)$$

Where:

- c represents the total number of groups in the corpus.
- And n_i^h is the number of documents from the h^{th} class that were assigned to cluster C_i [35].

In purity, it is expected that all the classes in each cluster be from the dominant class, by this; for a cluster to be optimal, the value of purity would be 1, which means the cluster contains only the documents from a specific category. Generally, the value of purity is a real number between [0, 1], the larger the purity value, the better the performance quality of the cluster is [35]. In contrast with purity, the lower the entropy value, the better the performance quality of the cluster is [35], the higher entropy value indicates the clustering performance is not as good as required to be. So, we expected that each cluster must have a high purity value and a low entropy value to maintain the quality needed for the clustering algorithm.

5.3. Experimental Settings

There are many sources of web documents, for our investigation we have used the following two real-world datasets, which are also used in the literature for the same purpose of semantic-based document clustering. For the purpose of diversity, we purposely selected two different datasets, which belong to different sources of knowledge.

5.3.1. BBC Dataset

Our experiments are on the real datasets of web documents. The first dataset used in our study is the BBC datasets, which holds the content of the original articles owned by the BBC News. It consists of 2,225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005. The topical class labels are [55]:

1. Business (510)
2. Entertainment (386)
3. Politics (417)
4. Sport (511)
5. Tech (401)

5.3.2. CMU World Wide Knowledge Base (Web->KB) Project

The *webkb* is a data set consisting of classified web pages. This data set is from the World Wide Knowledge Base project of the CMU text learning group, it contains www-pages collected from the departments of computer science of different universities in January 1997. The collected pages were 8,282 pages and were manually classified into the following categories [56]:

1. Student (1641)
2. Faculty (1124)
3. Staff (137)
4. Department (182)
5. Course (930)
6. Project (504)
7. Other (3764)

6. RESULTS

Table 6.1 and Table 6.2 show the purity and entropy results for semantic based clustering of web document corpora; BBC and Web->KB. Figure 6.1 and Figure 6.2 correspond to Table 6.1 and Table 6.2 respectively.

As Table 6.1 and Figure 6.1 demonstrate, on average for both input corpora, cosine similarity metric has the best performance of the overall for document clustering, then comes Euclidean and Pearson metrics of semantic similarity, and Feature_Overlap Coefficient has the weakest performance.

Table 6.1. Average purity results for both datasets

METRIC	AVERAGE PURITY
Cosine	0.427
Euclidean	0.426
Pearson Correlation Coefficient	0.425
Dice Coefficient	0.348
Jaccard Coefficient	0.347
Feature_Overlap Coefficient	0.344

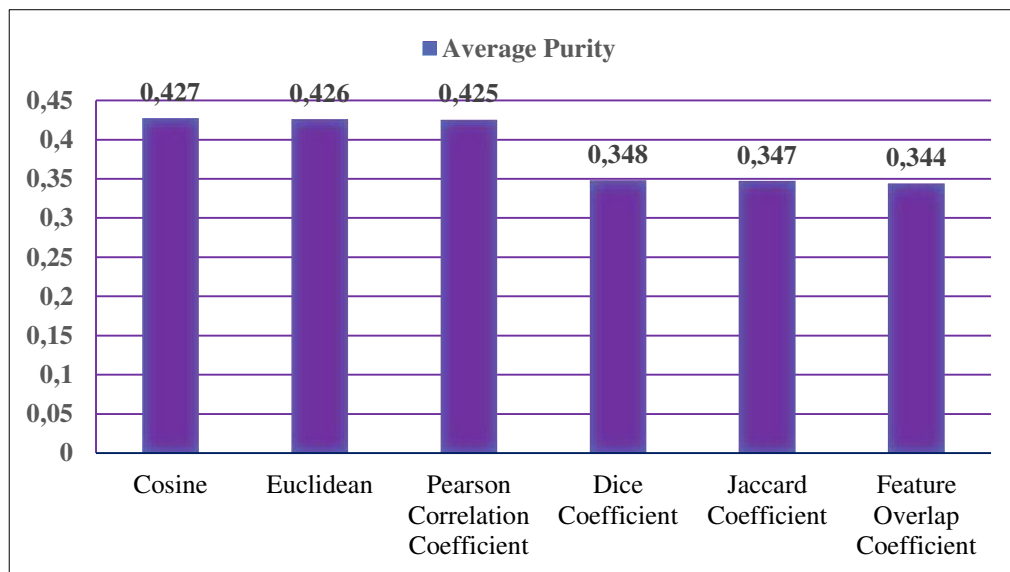


Figure 6.1. Average purity results for both datasets

Table 6.2 and Figure 6.2 demonstrates the average Entropy results for both input datasets BBC and Web->KB. Entropy measure is unlike the Purity measure, where the smaller Entropy values mean higher clustering quality.

As it is shown in Table 6.2 and Figure 6.2, again it is the Cosine metric, which has the best performance among the other five metrics of semantic similarity. Also, again come next after Cosine metric, both Euclidean and Pearson Correlation metrics. Jaccard showed the weakest performance in terms of Entropy metric.

Table 6.2. Average entropy results for both datasets

METRIC	AVERAGE ENTROPY
Cosine	0.77
Euclidean	0.78
Pearson Correlation Coefficient	0.79
Feature_Overlap Coefficient	0.87
Dice Coefficient	0.88
Jaccard Coefficient	0.89

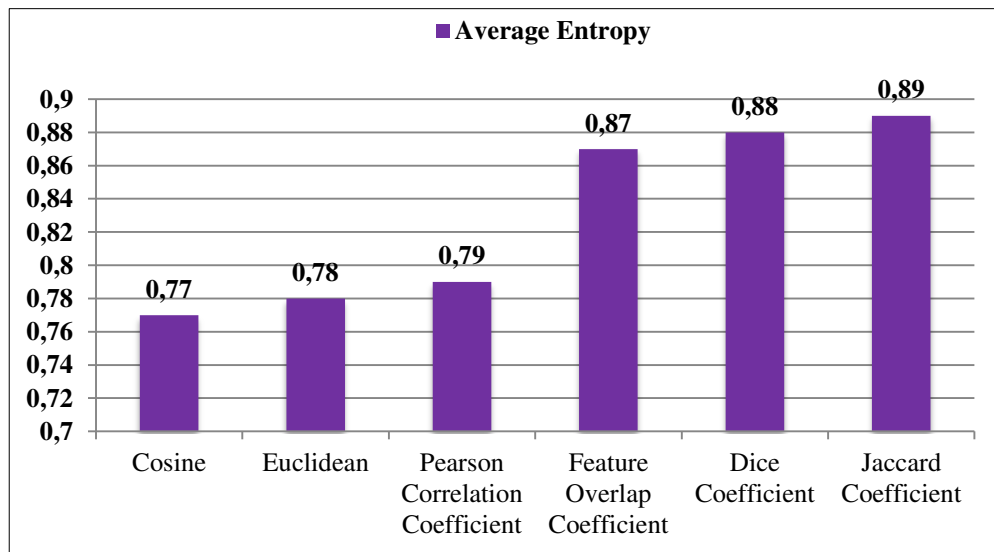


Figure 6.2. Average entropy results for both datasets

Most likely the reason for Cosine semantic similarity being the first metric for the purity and entropy metric is:

- Unlike tf-idf weighting schema, for latent semantics cosine semantic similarity metric computes the orientation of semantic relatedness among text documents. Also, unlike other similarity measures, the cosine similarity measures the direction length of meaning relatedness between text document feature vectors. As for given pair of document feature vectors (d_1, d_2) an angle near to 0° will give $\cos(0)$ value of 1, which means two documents d_1 and d_2 are semantically identical. Whereas, for a given pair of document feature vectors (d_1, d_2) an angle near to 90° will give $\cos(90)$ value of 0, which means two documents d_1 and d_2 are semantically unrelated. But for a given pair of document feature vectors (d_1, d_2) near to angle of 180° will give $\cos(180)$ value of -1, which means two documents d_1 and d_2 are semantically opposed, i.e. belonging to two different topics [34].

Our result also agree with the fact that Cosine measurement is also used to measure cohesion within clusters in the field of text mining such as:

- Search: finding the most similar documents to a given search query.
- Classification: are some customers likely to buy that product?
- Clustering: are there natural groups of similar documents?
- Product recommendations: which products are similar to customer's past purchases?

7. DISCUSSION AND CONCLUSIONS

It has been an open research problem of which semantic similarity/distance is best suitable with the LSA model for clustering the web documents. During our investigation study, we expected to find the best technique for semantic-based web document clustering with the LSA model. The results of our investigation study can help other researchers and industries as well on how to perform document clustering on semantic based methods, as there are many semantic similarity options that may be used with the LSA model.

To conclude, this investigation found the best use of LSA model in semantically clustering of the text documents. Among six possible variations implemented with the LSA model, and for the task of web document clustering, LSA model performed in different ways. On average, Cosine similarity was found as the best metric to be used with the LSA model for the task of web document clustering.

On average means the performance of the system evaluated together for both input datasets of BBC and Web->KB. Whereas, the Jaccard Coefficient metric has been found to give the weakest performance for the task of web document clustering. Also, both Pearson and Feature_Overlap metrics showed medium performance among all six metrics.

One more finding of this investigation is that, besides having many parameters in both LSA model and K-Means algorithm, the semantic similarity metric has a notable effect on the final outcomes of document clustering. A reasonable future work would be to investigate in depth other parameters included in the proposed system, such as the number of clusters in the K-Means algorithm and the number of dimensions the TF/IDF matrix is reduced to by the SVD technique. Also, possible future work could involve other text corpuses as inputs and involving other semantic similarity measurements.

REFERENCES

- [1] **Sathiyakumari, K., Manimekalai, G., Preamsudha, V., and Scholar, M. P.,** 2011. A Survey on Various Approaches in Document Clustering, *International Journal of computer technology and application (IJCTA)*, **2**, p. 1534–1539.
- [2] **Witten, I. H.,** 2004. Text Mining, p. 98.
- [3] **Martin, J. D.,** 1995. Clustering full text documents, *Proceedings of the IJCAI-95 Workshop on Data Engineering for Inductive Learning*, **95**, p. 1–10.
- [4] **Liu, Y., Zhang, D., Lu, G., and Ma, W. Y.,** 2007. A survey of content-based image retrieval with high-level semantics, *Pattern Recognition*, **40**, p. 262–282.
- [5] **Snoek, C. G. M. et al.,** 2009. The MediaMill TRECVID 2009 Semantic Video Search Engine, *TRECVID Workshop*.
- [6] **Antai, R., Fox, C., and Krudschwitz, U.,** 2011. The use of latent semantic indexing to cluster documents into their subject areas.
- [7] **Song, W. and Park, S. C.,** 2007. A novel document clustering model based on latent semantic analysis, Semantics, Knowledge, and Grid, *Third International Conference*, p. 539-542.
- [8] **Rott, M. and Cerva, P.,** 2014. Investigation of Latent Semantic Analysis for Clustering of Czech News Articles, *Database and Expert Systems Applications (DEXA), 2014 25th International Workshop*, p. 223-227.
- [9] **Jiaming, Z. and Loh, H. T.,** 2007. Using latent semantic indexing to improve the accuracy of document clustering, *Journal of Information & Knowledge Management*, **6**, p. 181-188.
- [10] **Kuta, M. and Kitowski, J.,** 2015. Comparison of Latent Semantic Analysis and Probabilistic Latent Semantic Analysis for Documents Clustering, *Computing and Informatics*, **33**, p. 652-666.

- [11] **Naik, M. P., Prajapati, H. B., and Dabhi, V. K.,** 2015. A survey on semantic document clustering, *Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference*, p. 1-10.
- [12] **Doan, T.,** 2003. Investigation on How to Improve Latent Semantic Analysis Performance, *Inquiry: The University of Arkansas Undergraduate Research Journal*, **4**, p, 22.
- [13] **Song, W. and Park, S. C.,** 2009. Genetic algorithm for text clustering based on latent semantic indexing, *Computers & Mathematics with Applications*, **57**, p. 1901-1907.
- [14] **Zheng, H. T., Borchert, C., and Kim, H. G.,** 2010. GOClonto: An ontological clustering approach for conceptualizing PubMed abstracts, *Journal of biomedical informatics*, **43**, p. 31-40.
- [15] **Marcus, A. and Maletic, J. I.,** 2003. Recovering documentation-to-source-code traceability links using latent semantic indexing, *Software Engineering, 2003 Proceedings. 25th International Conference*, p. 125-135.
- [16] **Gotoh, Y. and Renals, S.,** 1997. Document space models using latent semantic analysis.
- [17] **Bernard, L.,** 2017. what is the BNC? - About the British National Corpus, *Natcorp.ox.ac.uk*, <http://www.natcorp.ox.ac.uk/corpus/index.xml>
- [18] **Ozcan, R. and Aslangdogan, Y. A.,** 2004. Concept based information access using ontologies and latent semantic analysis, *Dept. of Computer Science and Engineering*, **8**.
- [19] **Wei, C. P., Yang, C. C., and Lin, C. M.,** 2008. A Latent Semantic Indexing-based approach to multilingual document clustering, *Decision Support Systems*, **45**, p. 606-620.
- [20] **Hasanzadeh, E., Rad, M. P., and Rokny, H. A.,** 2012. Text clustering on latent semantic indexing with particle swarm optimization (PSO) algorithm, *International Journal of Physical Sciences*, **7**, p. 16-120.

- [21] **Jain, A. K., Murty, M. N., and Flynn, P. J.**, 1999. Data clustering: a review, *ACM Computing Survey(CSUR)*, **31**, p. 264–323.
- [22] **Anusuya, M. A. and Katti, S. K.**, 2011. Classification Techniques used in Speech Recognition Applications: A Review, *Int. J. Comput. Technol. Appl.*, **2**, p. 910–954.
- [23] **Jain, A. K.**, 2010. Data clustering: 50 years beyond K-means, *Pattern Recognition Letters*, **31**, p. 651–666.
- [24] **Manning, C. D., Raghavan, P., and Schütze, H.**, 2009. Introduction to information retrieval, *Cambridge: Cambridge university press.*, **1**, p. 496.
- [25] **Chumwatana, T.**, 2014. Using Clustering Techniques for on-segmented Language Document Management : A Comparison of K-mean and Self Organizing Map Techniques, *Knowledge Management International Conference (KMICe)*, p. 600–600.
- [26] **Shah, N. and Mahajan, S.**, 2012. Document Clustering: A Detailed Review, *International Journal of Applied Information Systems*, **4**, p. 30–38.
- [27] **Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W.**, 2017. Scatter/gather: A cluster-based approach to browsing large document collections, *ACM SIGIR Forum*, **51**, p. 148-159.
- [28] **Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D.**, 2014. The Stanford CoreNLP Natural Language Processing Toolkit, *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, p. 55–60.
- [29] **Güngör, T.**, 2010. Part-of-Speech Tagging, p. 205-235.
- [30] LSA from Online Electronic Book from systems-sciences: <http://systems-sciences.uni-graz.at/etextbook/bigdata/lisa.html>

- [31] **Landauer, T. K. and Dumais, S. T.**, 1997. A solution to Plato's problem : The Latent Semantic Analysis Theory of Acquisition , Induction , and Representation of Knowledge, *Psychoogical Review*, **104**, p. 211–240.
- [32] **Choi, S. S., Cha, S. H., and Tappert, C. C.**, 2010. A Survey of Binary Similarity and Distance Measures, *Journal of Systemics, Cybernetics and Informatics*, **8**, p. 43–48.
- [33] **Cha, S. H.**, 2007. Comprehensive Survey on Distance / Similarity Measures between Probability Density Functions, *International Journal of Mathematical Models and Methods in Applied Sciences*, **1**, p. 300–307.
- [34] **Gomaa, W. H. and Fahmy, A. A.**, 2013. A Survey of Text Similarity Approaches, *International Journal Computer Applications*, **68**, p. 13–18.
- [35] **Huang, A.**, 2008. Similarity measures for text document clustering, *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, Christchurch, New Zealand, p. 49–56.
- [36] **Xu, R. and Wunsch, D.**, 2005. Survey of Clustering Algorithms, *IEEE Transactions on Neural Networks*, **16**, p. 645–678.
- [37] CS221-Handouts: K Means, written by Chris Piech, based on a handout by Andrew Ng.
<http://stanford.edu/~cpiech/cs221/handouts/kmeans.html>
- [38] **Sebastiani, F.**, 2002. Machine learning in automated text categorization, *ACM computing surveys (CSUR)*, **34**, p. 1-47.
- [39] **Sailaja, N. V., Padmasree, L., and Mangathayaru, N.**, 2016. Survey of Text Mining Techniques, Challenges and their Applications, *International Journal of Computer Applications*, **146**, p. 30–35.
- [40] **Gupta, V. and Lehal, G. S.**, 2009. A survey of text mining techniques and applications, *Journal of emerging technologies in web intelligence*, **1**, p. 60-76.
- [41] **Tan, A. H.**, 1999. Text Mining: The state of the art and the challenges, *Proceedings*

of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases, **8**, p. 65–70.

- [42] **Simoudis, E.**, 1996. Reality check for data mining, *IEEE Expert: Intelligent systems and their applications*, **11**, p. 26-33.
- [43] **Chen, C.L. P. and Zhang, C. Y.**, 2014. Data-intensive applications, challenges, techniques and technologies: A survey on big data, *Information Sciences*, **275**, p. 314–347.
- [44] **Thilagavathi, K. and Priya, V. S.**, 2014. A Survey on Text Mining Techniques, *International Journal of Research in Computer Applications and Robotics*, **2**, p. 41–50.
- [45] **Aggarwal, C. C. and Zhai, C.**, 2012. A survey of text clustering algorithms, In *Mining text data*, Springer US, p. 77-128.
- [46] **Talib, R., Hanif, M. K., Ayesha, S., and Fatima, F.**, 2016. Text Mining: Techniques, Applications and Issues, (*IJACSA*) *International Journal of Advanced Computer Science and Applications.*, **7**, p. 414–418.
- [47] **Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K.**, 2017. A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques, *arXiv:1707.02919*.
- [48] **Joachims, T.**, 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features, *Machine learning: ECML-98*, p. 137–142.
- [49] **Hotho, A., Nurnberger, A., and Paaß, G.**, 2005. A Brief Survey of Text Mining, *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, **20**, p. 19–62.
- [50] **Michelakis, E., Androutsopoulos, I., Paliouras, G., Sakkis, G., and Stamatopoulos, P.**, 2004. Filtron : A Learning-Based Anti-Spam Filter, *proceedings of the 1st conference on email and anti-spam (CEAS 2004)*.

- [51] **Witten, I. H., Don, K. J., Dewsnip, M., and Tablan, V.,** 2004. Text mining in a digital library, *Int J Digit Libr*, **4**, p. 56–59.
- [52] **Sharda, R. and Henry, M.,** 2009. Information extraction from interviews to obtain tacit knowledge: A text mining application, *AMCIS 2009 Proceedings*, p. 283.
- [53] **Dennis, S., Landauer, T., Kintsch, W., and Quesada, J.,** 2003. Introduction to latent semantic analysis, *Slides from the tutorial given at the 25th Annual Meeting of the Cognitive Science Society, Boston*.
- [54] **Kodratoff, Y.,** 1999. About knowledge discovery in texts: A definition and an example, *Unpublished Paper*, p. 1-16.
- [55] **BBC Datasets:** <http://mlg.ucd.ie/datasets/bbc.html>
- [56] CMU World Wide Knowledge Base (Web->KB) project:
<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/>

CURRICULUM VITAE

Mashhood Ali Ali

Email: sayedmashhood@gmail.com

phone: +90 506 170 1115

+964 750 431 0626

Education: Bachelor of Science, University of Duhok/Faculty of Science, Duhok /Kurdistan Region-Iraq, 2009-2013.

Work experiences:

- Assistant Programmer at Duhok Polytechnic University/Zakho Technical Institute, department of Information Technology, September 2013 - present.
- Database Management Officer for CfW Project for Summer Activities in Kurdistan Region of Iraq, Welthungerhilfe (WHH) NGO, November 2016 – December 2016.

Awards:

- Certificate of Participation in the **8th International Advanced Technology Symposium**, Elazig, Turkey on October 19-22-2017.
- Certificate of participation in the **Summer School Activities** Project as Community Volunteer - People in Need NGO, Zakho/Kurdistan Region-Iraq on August 14, October 25, 2016.
- Certificate of participating in the **International Cyber Security Workshop Certificate Program**, Gelişim University/Istanbul-Turkey, on May 23-27, 2016.
- Certificate of participating in the **Imagine Cup 2012** competition directed by Microsoft corporation, Erbil/ Kurdistan Region -Iraq on May 15-17, 2012 in.

Projects:

- Chrisk Kindergarten Database Management System.
- Human Resources Database Management System (HRDMS) for Directorate General of Health / Duhok-Kurdistan Region of Iraq (BSc. project).

- ECS+ (Educational Control System Plus) for monitoring every activity in the school and making relations among school and students and their parents (Imagine Cup 2012 project).

Languages:

- **Kurdish, Persian, English, Arabic, and Turkish.**

