Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Department of Software

Arabic Documents Clustering Using LDA2Vec for Topics Modeling

A Thesis

Submitted to the Council of the College of Information Technology for Postgraduate Studies of the University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology – Software

By

Doaa Wahhab Ibrahim Hadwan

Supervised By

Asst. Prof. Dr. Sura Zaki Naji Alwan

2022 A.D. 1444 A.H.

﴿ فُصِّلَتْ: 3﴾

Declaration

I hereby declare that this thesis entitled "**Arabic Documents Clustering Using LDA2Vec for Topics Modeling**" submitted to the University of Babylon in partial fulfillment of requirements for the degree of Master in Information Technology \ Software, has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for experts and summaries whose source is appropriately cited in the references.

Signature:

Name: Doaa Wahhab Ibrahim

Date: / /2022

Supervisor Certification

I certify that the thesis entitled (**Arabic Documents Clustering Using LDA2Vec for Topics Modeling**) was prepared under my supervision at the department of Software/ College of Information Technology/ University of Babylon as partial fulfillment of the requirements of the degree of Master of Philosophy in Information Technology-Software.

Signature:

Supervisor Name: **Asst. Prof. Dr. Sura Zaki Alrashid**

Date: / /2022

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "**Arabic Documents Clustering Using LDA2Vec for Topics Modeling**" for debate by the examination committee.

Signature:

Asst. Prof. Dr. Ahmed Saleem Abbas

Head of Software Department

Date: / /2022

Certification of the Examination Committee

We, the undersigned, certify that (**Doaa Wahhab Ibrahim**) candidate for the degree of Master in Information Technology - Software, has presented his thesis of the following title "**Arabic Documents Clustering Using LDA2Vec for Topics Modeling**" as it appears on the title page and front cover of the thesis that the said thesis is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:

Signature:
Name: **Dr. Yossra Hussain Ali**
Title: Assistant Professor
Date: / / 2022
(Chairman)

Signature:
Name: **Dr. Suad Abdullelah Abdulhussein**
Title: Assistant Professor
Date: / / 2022
(Member)

Signature:
Name: **Dr. Hussein Abdul Wasi Hussein**
Title: Lecturer
Date: / / 2022
(Member)

Signature:
Name: **Dr. Sura Zaki Alrashid**
Title: Assistant Professor
Date: / / 2022
(Member and Supervisor)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name**: Dr. Hussein Atiya Lafta**
Title: Professor
Date: / / 2022
(Dean of Collage of Information Technology)

Dedication

I dedicate this work with genuine gratitude and warm regard to…

The great martyrs who sacrificed their lives for the sake of this country

All children in this world

My great parents

who never stop giving of themselves in countless ways

My beloved brothers and sisters

who stand with me all the time without getting boring

My friends

who encourage and support me

Everyone who has touched my heart, helped me, or encouraged me from the beginning until now.

Acknowledgment

In the Name of Allah, the Most Merciful, the Most Compassionate All praise be to Allah, the Lord of the worlds; and prayers and peace be upon Mohamed His servant and messenger.

First of all, Praise Allah SWT for giving me the health and strength to complete this thesis.

I am thankful to my supervisor **Dr. Sura Zaki Alrashid**, without his help, guidance, and continuous follow-up; this research would never have been done.

Also, I would like to thank the academic staff of the Faculty of Information Technology/Software department at Babylon University who helped me during my Master's study and taught me different courses.

Last but not least, I am greatly grateful to my family for their continued love and support. Also, I would like to thank those who helped me with scientific information that benefited me in this research, thank you all.

Doaa Wahhab Ibrahim

May, 2022

Declaration Associated with this Thesis

Some of the works presented in this thesis have been published or accepted as listed below.

The published Paper:

Title: **A Topic Modeling for Clustering Arabic Documents**,

Author: Doaa Wahhab Ibrahim, Sura Zaki Alrashid

Journal: The 2nd International Conference on Information Technology to Enhance E-Learning and Other Applications - [2nd IT-ELA 2021]

Publisher: IEEE

Abstract

Due to the massive increase in the number of Arabic text documents available on the internet and in databases, researchers are facing a significant challenge in finding better methods to deal with a large amount of data. Therefore, it becomes necessary to develop effective techniques or tools to assist in the discovery and analysis of information in Arabic documents. Arabic document clustering is an important aspect of providing conjectural navigation and browsing techniques by organizing massive amounts of data into a small number of defined clusters.

In this thesis, an approach has been designed that employs the topic modeling as an Arabic document clustering technique. A recently developed topic modeling algorithm, LDA2Vec has been used in this approach. LDA2Vec is a hybrid algorithm introduced by Christopher Moody in 2016, that implements both words and topics into a single framework. LDA2Vec makes large amounts of text valuable to people (rather than machines) while making the model easy to modify. LDA2Vec results are a set of sparse document weight vectors, as well as easily interpretable topic vectors.

The approach consists of several stages which are collecting text documents, text pre-processing, text representation, training stage using the LDA2Vec algorithm, testing stage, and evaluation of the model. The developed approach has been tested using an Arabic news dataset used in previous similar studies. The results showed that the LDA2Vec model is superior in terms of clustering quality for Arabic text documents according to external measures like purity, F-measure, accuracy, and other measures. It is shown in this thesis that the purity of the developed approach is 0.88 compared to 0.75 for Latent Dirichlet Allocation (LDA), one of the most widely used topic modeling techniques, these results are higher in comparison to a recent similar study.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
BOW	Bag of Words
CBOW	Continuous Bag of Words
DL	Deep Learning
DM	Data Mining
FN	False Negative
FP	False Positive
IR	Information Retrieval
ISRI	Information Science Research Institute
LDA	Latent Dirichlet Allocation
LEs	Learning Embeddings
ML	Machine Learning
NLP	Natural Language Processing
NMI	Normalized Mutual Information
OSAC	Open Source Arabic Corpora
SG	Skip Gram
SGNS	Skip Gram Negative Sampling
SVM	Support Vector Machine
TC	Text Clustering
TF-IDF	Term Frequency-Inverse Document Frequency
TN	True Negative
TM	Text Mining
TP	True Positive
WEs	Word Embeddings

List of Algorithms

CHAPTER ONE

INTRODUCTION

1.1 General Introduction

In recent years, the internet is full of information and knowledge sources and the most of information on the Internet has taken the form of texts [1]. The Number of Arabic documents on the Internet is rapidly increasing. Finding relevant Arabic documents has become increasingly intriguing to Arabic users [2]. The massive increase in the number and availability of online text documents might be confusing to readers, forcing them to spend time and effort searching for appropriate information on specific topics of interest [3].

Retrieving essential information from a huge collection is a difficult task because the data is contained in text documents. This could be handled by the "information retrieval process", which uses a parallel search approach to retrieve text documents. As a consequence, developing efficient approaches or tools to aid in the discovery and analysis of information in documents became essential [4].

Text mining, commonly known as text analysis, is a procedure for determining what events or concepts are being discussed in a document. Text mining converts unstructured text into structured data that can be analyzed easily using various techniques such as (information extraction, categorization, clustering, etc.). Text mining uses natural language processing (NLP), which allows machines to understand and process human language automatically because this information is clear to a person who reads a document, but a program is provided only with the text, as it is written not the topic of each document. In order to perform this task in a program, data scientists apply an approach called topic modeling [5].

Topic modeling is a probabilistic approach and a form of text mining used to discover the abstract topics in a collection of documents by

presenting the document as a probability distribution over topics and a topic as a probability distribution over a words dictionary [6]. Topic modeling and clustering are both unsupervised learning approaches that need a number of topics/clusters to be given in advance and do not require labels to work. Topic modeling has gotten a lot of attention in the field of study in the last few years. It is employed in many applications, including information retrieval (IR) and natural language processing (NLP) [7].

LDA is one of the most widely used topic modeling approaches. LDA has been frequently utilized to discover hidden (latent) semantic topical structures in a huge corpus of documents [8]. Topic modeling techniques are used in NLP to represent (or embed) words as vectors in a continuous, multidimensional vector space with important geometrical correlations between word vectors. A common approach that uses neural networks to create vector spaces for NLP called Word2vec [9]. The combination of these two techniques is known as LDA2Vec.

LDA is a technique to describe a global relationship among documents, while word2vec predicts words in a local manner, and LDA2Vec combines these features of LDA and Word2vec.

Additionally, The Arabic language is the 5th most frequently spoken language in the world. Around 300 million people speak Arabic as their first language, with another 250 million using it as a second language [10]. The Arabic documents have become very popular on the Internet in last few years. The Arabic language is one of the most diverse human languages in terms of sentence structures and meaning, and owing to the language's unique morphological principles, there are limited works in the literature on the retrieval or mining of Arabic electronic text documents [11].

1.2 Motivation and Problem Statement

The amount of data on the internet keeps increasing every single day and a considerable portion of such data is unstructured text data, which forms about (80%) of the world's data. Retrieving and searching for information has become a difficult task and a challenge for web users. Arabic documents have been increasingly common in electronic format in recent years, whereas English has been employed as the primary language in most research on various information retrieval tasks (such as classification, clustering, and search). As a result, it becomes necessary to create efficient techniques or tools to assist in discovering and analyzing content in Arabic documents.

1.3 Aim and Objectives of Thesis

This thesis aims to cluster Arabic documents using a hybrid technique of topic modeling called (LDA2Vec), which is a combination of two methods, LDA and Word2vec. To accomplish the thesis aim, various objectives will be discussed, including:

- Pre-processing the found documents, such as removing Arabic stop words, Arabic diacritics, numbers, special characters, and non-Arabic letters, and normalizing specific Arabic letters. This may reduce text size, enhance text processing, and eliminate extraneous words.
- Study how different Arabic text document representation techniques, such as a bag of words (BOW) with building word vectors from scratch and using pre-trained word embedding, affect the results of the proposed approach.

- Investigate the impact of changing the values of the parameters that influence word selection and distribution on the clustering results.

1.4 Thesis Significance and Contributions

The major contributions of this thesis are:

1. Proposing an approach for clustering Arabic text documents using a deep learning topic modeling technique that learns the word, document, and topic vectors concurrently in order to predict document topics.
2. A Procedure for unseen document testing to discover document topics with LDA2Vec algorithm is suggested in this thesis work.

The significance of the current thesis study originates from the creation of Arabic word vectors in order to generate semantically related word clusters with high similarity. These clusters may outperform "generic" word vectors in Arabic semantic applications such as question answering systems, search engines, and query expansion.

1.5 Thesis Challenges

The most significant challenges in this research are:

1. The Arabic language, with its rich characteristics and unique morphology, is a major challenge for IR systems. As a result, there are relatively few studies in the literature about retrieving or mining Arabic electronic text documents.
2. Arabic language uses various meanings for the same words and makes use of diacritics to reflect word meaning. As a result, standard methods of stemming and normalization are inadequate.

3. Reducing the number of dimensions in data without missing critical information.
4. How to design an effective model for clustering Arabic text that can improve clustering performance?

1.6 Related Works

LDA2Vec is a new approach proposed by Christopher Moody and much research related to this approach is still not done. To the best of the researcher's knowledge, this is the first effort to review the implementation of the LDA2Vec approach for clustering Arabic documents. In contrast, the studies related to using LDA2Vec applied to English documents and some other languages.

1.6.1 Clustering and Topic Modeling

Kelaiaia and Merouani (2016) [12] in their study, compared LDA with K-means to see how LDA reacts to clustering Arabic texts, which is a very flexible language. The experiment was carried out on a collection of Arabic documents that serve as a benchmark (OSAC, Open Source Arabic Corpus). F-measure and Entropy were the two metrics used for evaluation. The results consistently demonstrated a clear improvement of LDA over K-means on raw, cleaned, and stemmed forms of the document collection.

In [13] Wang et.al, (2016) suggested a hybrid approach for extracting features from documents in a semantic space with bag-of-distances. The hybrid approach, which employs both Word2vec and LDA, not only produces relationships between documents and topics but also integrates contextual relationships among words. The suggested method was tested using the Twenty Newsgroup dataset and assessed using the f1-score metric. The experimental results showed that the hybrid method's document features were effective for improving model performance by

consolidating both global and local relationships. It has been demonstrated that the suggested method outperforms three alternative single models.

Esposito et.al, (2016) [14] this study aimed to evaluate and compare two different frameworks for unsupervised topic modeling techniques (LDA and Word2vec) in the Computational White House Press Briefings (CompWHoB) Corpus, a political corpus that collects transcripts from the American Presidency Project website.

Two experiments were conducted: the first has used a traditional LDA model, while the second has used Word2vec to train the model and create word embedding for the topic modeling test set. After clustering the output of the two techniques, they were evaluated using the purity criterion. It was shown that the purity of LDA is 0.46 compared to 0.54 for Word2vec, but only when a linguistic task-oriented preprocessing stage is being used.

Alhawarat and Hegazi (2018) [7] used a hybrid approach to cluster Arabic text documents in their work. They primarily employed generative models and clustering methods. In this research, the LDA and k-means clustering algorithms were applied to a news dataset that has previously been used in comparable studies. External evaluation metrics such as purity, F-measure, accuracy, Jaccard index, and others reveal that the combined method exceeds previous methods in terms of clustering quality for Arabic documents. The purity of the combined approach was higher in this research compared to the k-means algorithm, and these results were better than in a comparable identical study.

1.6.2 LDA2Vec

In [15] **C.Moody (2016)** demonstrated how to build unsupervised document representations that provide coherent topics using a simple model called LDA2Vec, this is achieved by extending skip-gram negative

sampling (SGNS). Word, topic, and document vectors were all simultaneously trained and embedded in the same representation space that preserves semantic regularities between learned word vectors while still producing sparse and interpretable document-to-topic proportions in the LDA approach.

He utilized the Twenty Newsgroups and Hacker News comment corpus to test the model. As a result, the model was simple to apply in automated differentiation frameworks and can lead to unsupervised representations that are more easily interpretable. He did not give processing for testing new documents in his study.

Xue Mu (2019) [16] proposed a text feature representation model that combines the LDA and the Word2Vec models and then used it to solve the text retrieval problem. The proposed method calculated the distance between documents and topics, then represented each document as a feature vector with each dimension denoting the distance between this document and a specific topic. To evaluate the performance of the proposed algorithm, several related methods were applied, and experimental results on the Twenty Newsgroups dataset proved that the proposed solution outperforms other methods and can achieve high text retrieval accuracy.

Luo and Shi (2019) [17] suggested an approach for automatically identifying more interpretable narrative text topics in aviation safety reports. By combining the word vector and the document topic vector into the same vector space, the LDA2Vec model was utilized to train the document-topic probability distribution matrix and the topic-word probability distribution matrix. The topic was manually analyzed and recognized, and the metadata from the topic and the report were combined

to provide the desired output results. When it comes to detect the latent topic of narrative text in aviation safety reports, the experimental results reveal that the proposed topic identification approach is more interpretable than the LDA model.

Hasan et al. (2019) [18] conducted an experiment on Bangla language to evaluate the performance of two topic modeling approaches, LDA and LDA2Vec. This study aimed to find a method for automatically analyzing and categorizing Bangla news documents. For testing and implementation purposes, they developed a technique to discover topics for non-factorized documents from LDA and LDA2Vec. According to their findings, LDA2Vec achieved 85.66 % over topics for test documents, compared to 62.45 % for LDA.

Mishra et al, (2020) [19] In this study, Latent Dirichlet allocation (LDA) and the LDA2Vec models were used for sentiment classification. The performance of both models was evaluated using a corpus of 1000 records. After running the models, the results proved that the hybrid technique of LDA2Vec (LDA + Word2Vec) provides superior accuracy to the traditional LDA.

In [20] **Culmer and Uhlmann (2021)** compared the performance of LDA2Vec combined with temporal tweet pooling (LDA2VecTTP) to traditional LDA and the Biterm Topic Model (Biterm), which was created specifically for topic modeling on short text documents. They used three different tweet pooling techniques for each of the three topic modeling algorithms: no pooling, author-based pooling, and temporal pooling. They then used each of the algorithms and tweet pooling strategies to perform topic modeling on two Twitter datasets. The findings from the biggest

dataset demonstrate that LDA2VecTTP can provide higher coherence scores as well as more logically coherent and interpretable topics.

Table 1-1 Summary of Related Works

	No.	Ref.	Authors	Year	Dataset	Methods	Evaluation Metrics	Results
Clustering and Topic modeling	1.	[12]	Kelaiaia, Merouani	2016	OSAC	K-means LDA	F-measure Entropy	21,96% 4,44%
	2.	[13]	Wang, Long Ma, Zhang	2016	Twenty Newsgroups	TF-IDF+SVM	F1-score	0.822
						Word2Vec+SVM		0.717
						LDA +SVM		0.639
						LDA+Word2vec		0.803
	3.	[14]	Esposito, Corazza, Cutugno	2016	CompWHoB	LDA	Purity	0.46
						Word2vec		0.54
	4.	[7]	Alhawarat, Hegazi	2018	Arabic News dataset	LDA	Purity Accuracy	0.74 0.78
						LDA + K-means		0.88 0.95
LDA2Vec	5.	[21]	Christopher Moody	2016	Twenty Newsgroups	LDA2Vec	Coherence	0.567
	6.	[16]	Xue Mu	2019	Twenty Newsgroups	TF-IDF	F1-score	0.652
						LDA		0.729
						Word2vec		0.803
						LDA+ Word2vec		0.814
	7.	[17]	Luo, Shi	2019	ASRS	LDA	Coherence	0.562
						LDA2Vec		0.588
	8.	[18]	Hasan, Hossain, Ahmed, Rahman	2019	Bangla News Corpus	LDA	Accuracy	62.45%
						LDA2Vec		85.66%

	9.	[19]	Mishra, Rajnish, Kumar	2020	1000 Reviews Corpus	LDA	Precision Recall	0.54 0.44
						LDA2Vec		0.82 0.78
	10.	[20]	Culmer, Uhlmann	2021	Twitter datasets #AllLivesMatter	LDA	**Coherence** • C_p • C_{UCI} • C_{UMass} • C_{NPMI}	-0.293 -0.752 -1.161 -0.027
						Biterm		-0.293 -0.959 -1.641 -0.026
						LDA2Vec		-0.388 -1.155 -2.374 -0.037

As previously stated, the past studies that dealt with the LDA2Vec algorithm did not address the performance of the algorithm with Arabic texts. As a consequence, the current work will concentrate on verifying the algorithm's performance with Arabic documents within a specific approach that consists of a number of stages for processing Arabic texts, as well as employing pre-trained word vectors to optimize the efficiency of the proposed approach.

1.7 Thesis Organization

The remainder of this thesis work is organized as follows:

- **Chapter Two**: is entitled "**Theoretical Background**". This Chapter provides an extensive description of text mining concepts and methodologies, text clustering and evaluation methods, topic modeling concept, and topic modeling algorithms.
- **Chapter Three**: is entitled "**The Proposed Approach**". It covers the proposed approach and its algorithm.
- **Chapter Four**: is entitled "**Experimental Results and Discussion**". This chapter demonstrates the results of the proposed approach and the research experiments. It also discusses the evaluation of the approach's performance.
- **Chapter Five**: is entitled "**Conclusions and Future Works**". This chapter presents the research conclusions and the possible future research directions to improve this work.

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 Introduction

This chapter investigates text mining and topic modeling approaches. It provides an overview of text mining, followed by a discussion of text mining methodologies such as Natural Language Processing (NLP) and text clustering, and clustering evaluation methods. For topic modeling, the context of topic modeling will be explained, including the utilized implementation techniques; LDA, and the used algorithm (LDA2Vec).

Since the thesis focuses on the analysis of Arabic documents. First, a brief introduction to the Arabic language will be introduced and its morphological principles will be demonstrated.

2.2 Arabic Language

Arabic language is one of the five most frequently spoken languages in the world. It is the official language of around 25 Middle Eastern and North African countries. Around 422 million people speak it as a first or second language. Because of the rising use of Arabic language on the web and in social media, natural language processing in Arabic language has recently gotten a lot of attention. Arabic language comes in a variety of forms and dialects, depending on the country or area. Arabic language in its formal form is utilized for formal purposes such as education and news [22], [23].

The Arabic language utilizes different letters for writing texts, documents, and scripts. The Arabic alphabets or letters are made up of the following 28 characters, they all represent consonants [24].

أ ب ت ث ج ح خ د ذ ر ز س ش ص

ض ط ظ ع غ ف ق ك ل م ن هـ و ي

Besides the Arabic letter hamza (ء) is regarded a perfect alphabetical letter according to some linguists. Unlike the English language, the Arabic language is written from right to left in a cursive style.

In certain contexts, the letters Alif (أ), waw (و), and ya (ي) may also be used to denote long vowels. In the Arabic script, short vowels are expressed by diacritical markings called (Harakat / حَرَكات), which are placed above or below regular consonant letters. These short vowels are the fatHa (الفَتْحَة), the kasra (الكَسْرَة), and the DHamma (الضَّمَّة). There are two further significant marks: the sukoon (السُّكُون) and the shaddah (الشَّدَّة). A less common diacritical mark is the tanween (التَّنْوِيْن). They are used to ensure that words are pronounced correctly [25].

Table 0-1 Arabic Diacritical Marks

Arabic Diacritical Marks				
Tanween with shaddah	Tanween تَّنْوِيْن	Short vowels with shaddah شَدَّة	Short vowels	
ـًّ	ـً	ـَّ	ـَ	fatHa فَتْحَة
ـٍّ	ـٍ	ـِّ	ـِ	kasra كَسْرَة
ـٌّ	ـٌ	ـُّ	ـُ	DHammah ضَمَّة
		ـّ	ـْ	sukoon سُكُون

2.3 Arabic language challenges

Arabic language is a challenging language for many reasons [26]:

- **First**: in Arabic language, orthographic variants are common; some combinations of letters may be written in different ways.

- **Second**: the morphology of Arabic language is very complicated.
- **Third**: the use of broken plurals is also common. Broken plurals are comparable to irregular English plurals in that they do not necessarily match the singular form as well. Existing stemmers do not handle broken plurals since they do not follow standard morphological rules.
- **Fourth**: Arabic words are typically confusing owing to the trilateral root structure. A word in Arabic is usually derived from a root, which usually consists of three letters. One or more of the root letters may be omitted in certain derivations, making many Arabic words exceedingly confusing with each other.
- **Fifth**: in written Arabic language, short vowels are eliminated.
- **Sixth**: synonyms are often used, maybe because Arabic speakers value variation in language as part of a good writing style.

As previously stated, the Arabic language is very challenging and, consequently, provides different opportunities for research and investigation in many fields, including artificial intelligence domains such as machine learning (ML), deep learning (DL), and natural language processing (NLP), among others.

2.4 Text Mining (TM)

Text mining, or known as "Text analysis," is an artificial intelligence technology that is used to extract meaningful and relevant models for knowledge exploration from huge textual data sources [27]. There are several sources of text data including digital libraries, news outlets, and other textual information such as social networks, articles, blogs, e-mails,

etc. with the majority of this text data being semi-structured or unstructured [28].

Text Mining is a multidisciplinary field that encompasses and integrates the methods of data mining, information retrieval, statistics, machine learning, and computational linguistics. There are several text mining techniques, such as summarization, classification, clustering, and topic modeling [29].

One of the most essential methodologies that text mining utilizes to handle text is Natural Language Processing (NLP), that enables machines to interpret and process human language automatically.

2.5 Natural Language Processing (NLP)

NLP is a sub-field of artificial intelligence that allows machines/computers to interpret human language. NLP examines the grammatical structure of sentences as well as the specific meanings of words before using algorithms to extract meaning and generate outputs. [30]. Most NLP approaches rely on machine learning to extract meaning from human languages. Furthermore, natural language processing is a subfield of linguistics and computer science. Linguistics is the scientific study of language, encompassing its structure, grammar, meaning, and phonetics. The study of linguistics or natural language processing is one of the most rapidly and far-reaching new technologies in computer science that covers overlapping subject areas such as Machine Learning (ML), Deep Learning (DL), and Artificial Intelligence (AI) [31]. Figure 2.1 demonstrates the coherence of these fields of study as a Venn diagram.

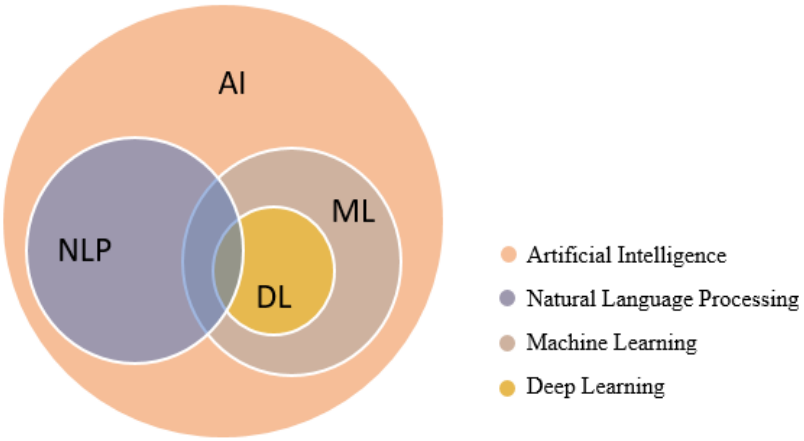


Figure 2-1 Venn diagram for natural language processing

2.6 Text Clustering (TC)

Text clustering is an important stage in text data analysis that has gotten a lot of attention from the text mining community. Clustering models attempt to categorize objects in a valid form (clusters) based on their similarities. It uses NLP and machine learning to understand and categorize (unstructured) textual data [32]. Web search engines, for example, use clustering to find relevant documents to a given query among a collection of lists with similar documents [33].

Text clustering (also known as Document clustering) is a technique for determining the similarity of two documents by examining a large number of textual features. It is widely used in areas such as topic extraction, and information retrieval. However, the difficulty is continuous because documents have a variety of features, and statistical similarity between documents does not always reflect semantic similarity. In many situations, two documents referring to two completely distinct topics will have a substantial number of words in common [34].

2.7 Handling Unstructured Data – Text Pre-Processing

Data that is structured is stored in a fixed field in a record or file. This data is kept in spreadsheets and relational databases. Unstructured data, on the other hand, is the opposite of structured data and refers to data that does not exist in a classical row-column database. Semi-structured data is data that is neither raw nor structured in a traditional database system [35].

Preprocessing procedure is the initial phase that plays a very essential part in text mining methods and applications. Due to the Arabic language's richness, which includes complicated morphology as compared to other languages. Preprocessing techniques have been used to improve the performance of Arabic text mining.

2.7.1 Tokenization

Tokenization is the process of breaking the text into smaller meaningful portions called tokens, and it is a highly significant stage in natural language processing (NLP). In the current thesis, tokenization was conducted by separating the text documents into words depending on the spaces between them [36].

2.7.2 Stop Words Removal

The stop words are the most frequent terms in any language (like articles, pronouns, conjunctions, prepositions, etc.) and do not offer much information to the text [37]. By removing these frequently occurring words from indices, the number of words against which each search term must be matched is reduced, resulting in a considerable improvement in query response time without compromising accuracy. Stop Words in Arabic language are any words that are not considered part of speech, i.e., nouns or verbs (including pronouns such as (هو, هي, هم), prepositions such as(من

, في , إلى), demonstratives such as (هذا , هذه , هؤلاء) adverbs such as (فوق , تحت), and so on) [38].

2.7.3 Text Cleaning and Normalization

Normalization seeks to reduce some letters in a word that have several forms to a single form. It is carried out in the manner outlined below [39].

- Removing Arabic short vowels (diacritics) (ّ، ُ، ً، َ، ِ، ٍ، ْ)
- Replacing ة by ه
- Replacing آ , ااا , إ , أ by ا
- Replacing congestive set of spaces and new line characters by single space, so that each two words will be separated by one space.

Moreover, the text is cleaned by deleting numerals, non-Arabic letters, single Arabic letters (letters that are not part of a word), punctuations, special characters.

2.7.4 Arabic Stemming

Stemming is a technique that lowers inflection in words to find the original root or stem of words by getting rid of any additional prefixes and suffixes in words, hence aiding in the preprocessing of text, words, and documents for text normalization. Arabic words have more complex forms in any other language with similar additions. Changes in the forms of words morphology have the same relevance as changes in the meaning of words and may be regarded as comparable to what is meant in retrieving information systems [40].

Arabic stemming approaches are categorized into two types based on the level of analysis desired: the root-based stemming approach and the light stemming approach.

a. **Root-based stemming approach** applies morphological approaches to detect the root. Many algorithms have been created for this strategy, which first peels off layers of suffixes and prefixes before checking a list of formats and roots to see whether the remains are a known root with a known format. If this is the case, it returns the root. Otherwise, it returns the original, unaltered word. The Information Science Research Institute's (ISRI) stemmer is one of the famous root-based stemmers, which is considered a rule-based stemmer that stems the word according to particular rules to locate its root. It is similar to Khoja stemmer but does not require a root dictionary [41].
b. **The light stemming approach** is the process of removing the most common suffixes and prefixes from a given list of prefixes and suffixes. Because the light stemmer approach does not attempt to extract the root of an Arabic word, it avoids infixes and does not discover patterns [38]. For the Arabic language, Several light stemmers have been suggested such as Al-stem by Darwish and Oard [42].

2.8 Handling Unstructured Data - Text Representation

Text representation is one of the most essential steps in data mining (DM), text mining (TM), information retrieval (IR), and Natural Language Processing (NLP) approaches. It seeks to numerically represent the unstructured text documents in order to make them mathematically computed [43]. This process, although repetitive, plays a critical role in selecting features of the machine-learning model.

2.8.1 Documents Representation

Document representation aims to represent document input into a fixed-length vector, which could describe the contents of the document, to reduce the complexity of the documents and make them easier to handle [44]. One of the simplest models to represent documents is a bag of words.

- **Bag of Words Model (BOW)**

  The bag of words (BOW) is an NLP of text modeling. BOW is easily understandable and easy to implement, and it has shown to be effective at tasks like language modeling and document categorization. A vector space model is another name for the BOW document representation model. The main purpose of the vector space model is to turn each variable-length document into a fixed-length vector. This vector represents the frequency of all words in the document. Every row represents an observation, and each feature represents a unique word [45].

2.8.2 Words Representation (word embeddings)

Most natural language processing applications required a word representation stage, which is a sort of learned representation which enables similar meaning words to have the same representation. Hence, many approaches to representing words as dense vectors in a low-dimensional vector space have been developed, each adopting a different training strategy inspired by neural network language modeling [45]. Word2vec and Global Vectors (GloVe) are two effective deep learning approaches for word embeddings [46]. In the current work, Word2vec that is used for learning word embeddings will be discussed.

- **Word2vec Model**

Word2vec model is a word representation model created at Google in 2013 by Tomas Mikolov [47]. This model employs two hidden layers in a shallow neural network to generate a vector for each word. Word vectors could be obtained using two methods: Continuous Bag of Words (CBOW) and Skip Gram (SG) models. In order to get a better representation of words, it is recommended to train the corpus using the huge corpus. Word2Vec has shown to be effective in a wide range of Natural Language Processing (NLP) tasks [48].

1. Continuous Bag of Words (CBOW) Model

The context for a particular target word is provided by surrounding words in the CBOW model as shown in Figure 2.2. The word representation is built by maximizing the (log-) probability of predicting the target word given its context. The CBOW model has a simplistic neural architecture in which the nonlinear hidden layer is eliminated and the projection layer is shared by all words [49]. The model optimizes the following for a given target word w_t and its context $\{w_{t-c},\ldots,w_{t+1},\ldots,w_{t+c}$

$$\frac{1}{|v|}\sum_{t=1}^{|v|}\log[P(w_t|\,w_{t-c},\ldots,w_{t-1},w_{t+1},\ldots,w_{t+c})] \qquad (1)$$

Where $|v|$ donates the number of words in the corpus and c donates the size of the dynamic context of w_t.

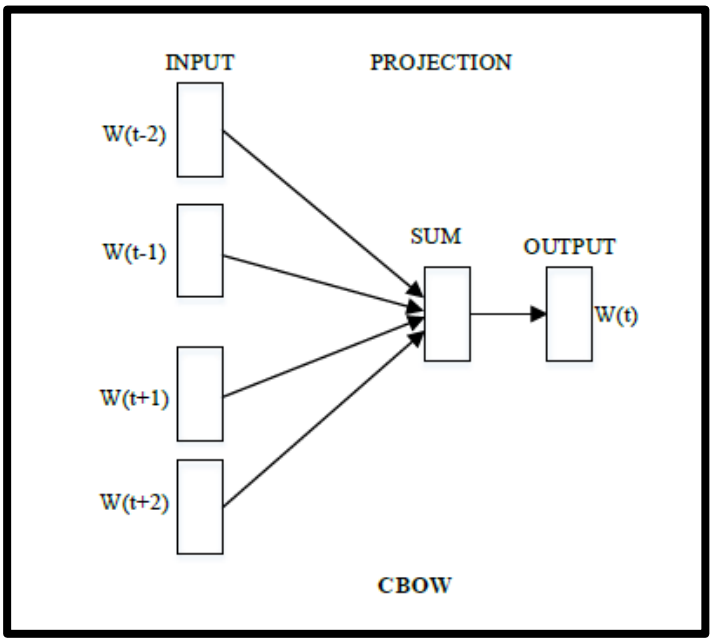


Figure 2-2 CBOW Model [50]

2. Skip-Gram (SG) Model

In contrast to the CBOW model, the skip-gram (SG) model calculates the current word using context words as shown in Figure 2.3. It has a similar architecture in that the neural network's input and output are reversed [50]. Each word vector in a corpus is trained to maximize the (log-) probability of creating neighboring words. Given a set of training words $\{w_{t-c},\ldots,w_{t+c}\}$, the model maximizes the average (log) probability of predicting the context of the current target word [49]:

$$\frac{1}{|v|}\sum_{t=1}^{|v|}\sum_{j=t-c,j\neq t}^{t+c}\log[P(w_j|w_t)] \qquad (2)$$

Where $|v|$ donates the number of words in the corpus and c donates the size of the dynamic context of w_t.

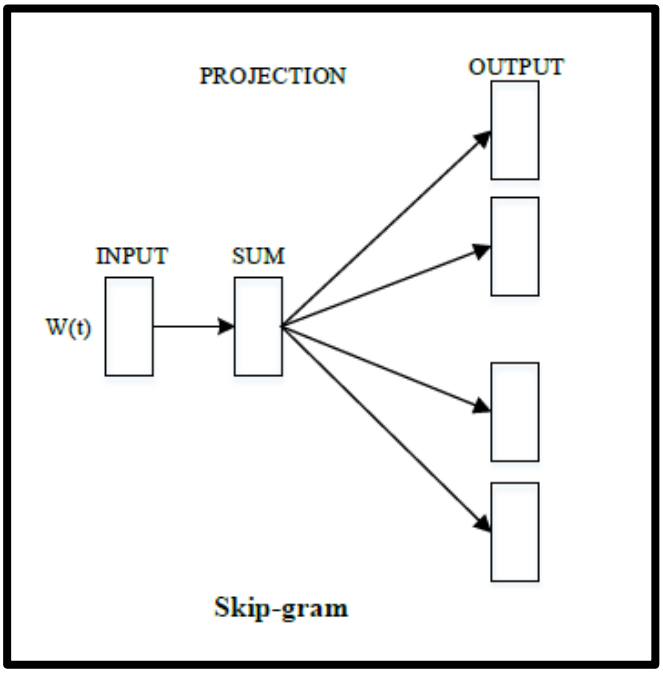


Figure 2-3 Skip-Gram Model [50]

- **Skip-Gram Negative Sampling (SGNS) Model**

Skip-Gram Negative Sampling seeks to maximize the similarity of words when they appear in the same context and minimize it when they occur in different contexts. SGNS trains a neural network to calculate the probability of encountering a context word c given a target word w sliding symmetric window throughout a subsampled training corpus with uniformly sampled window size from the range [1,w_{in}]. Each observed (w, c) pair is combined with a random selection of words ($2 \leq k \leq 20$) based on the training size, and the loss function is computed [51].

$$L_{wc}^{SGNS} = \log \sigma(W_w \widetilde{W}_c^{\,T}) + \sum_{i=1}^{k} E_{wi \sim P_n(w)} \log \sigma(- W_w \, \widetilde{W}_{wi}^{\,T}) \quad (3)$$

Where $P_n(w)$ is the distribution from which the noise words w_i are sampled, and more frequent words are more likely to be chosen as negative samples.

- **Pre-trained Embeddings**

Word Embedding is a powerful deep learning approach for creating words and documents vector representations. For training and generating an appropriate vector for each word, Word2vec requires very large corpora. Google, for example, utilize around 100 billion words to train Word2vec algorithms then re-released pre-trained word vectors with 300 dimensions [46]. Hence, Pre-trained Word Embeddings are embeddings learnt in one task and applied to another comparable task. These embeddings are trained on large datasets, stored, and then applied to different tasks [52].

2.9 Topic Modeling

Nowadays, topic modeling has received a lot of attention in the area of research. It is utilized in a variety of fields, most particularly in information retrieval and natural language processing. The goal of topic modeling is to discover a predefined number of topics. Topic modeling is an unsupervised technique based on statistical ideas that require no prior knowledge of the data. It has the ability to greatly assist users in understanding text corpora [53].

Clustering and topic modeling are quite similar in that they both need the number of categories to be defined ahead of time and do not require labels to operate. Using probabilistic approaches, topic models extract a collection of topics, and each topic contains a set of words. They extract a number of topics from a text corpus, with each topic characterized as a statistical distribution of a set of words. This is done via the use of probabilistic models [53].

One of the most common topic modeling techniques (LDA) and the most recent topic modeling technique (LDA2Vec) linked to approaches

addressed in information retrieval and machine learning literature are as follows.

2.9.1 Latent Dirichlet Allocation (LDA)

The Latent Dirichlet Allocation (LDA) is the most popular topic modeling technique, which is a probabilistic obstetrical model used to represent any combination of separated data. The purpose of LDA is to determine which topics a document belongs to, based on the words in it [54].

LDA is a three-layer Bayesian probability model and a comprehensive document generating model. Figure 2.4 illustrates a graphical model representation of LDA based on [55]. The observable variable is denoted by the shadow circle w, whereas the other latent variables are denoted by non-shadow circles. The boxes are "plates" representing the replicates. The outer plate denotes the documents, while the inner plate denotes the recurrent selection of latent topics and words inside a document. The document-topic distribution is represented by θ, and each is derived separately from the symmetric Dirichlet prior α. The topic-word distribution is designated as φ, and each is derived from a symmetric Dirichlet prior β. Based on these concepts and notations, LDA assumes the following generating process for each document in the corpus [56].

1. Choose $N \sim$ Poisson (ξ).
2. Choose $\theta \sim$ Dir (α).
3. For each of the N words w_n :
    a. Choose a topic $z_n \sim$ Multinomial (θ).
    b. Choose a word w_n from $p(w_n|z_n,\beta)$, a multinomial probability based on topic z_n.

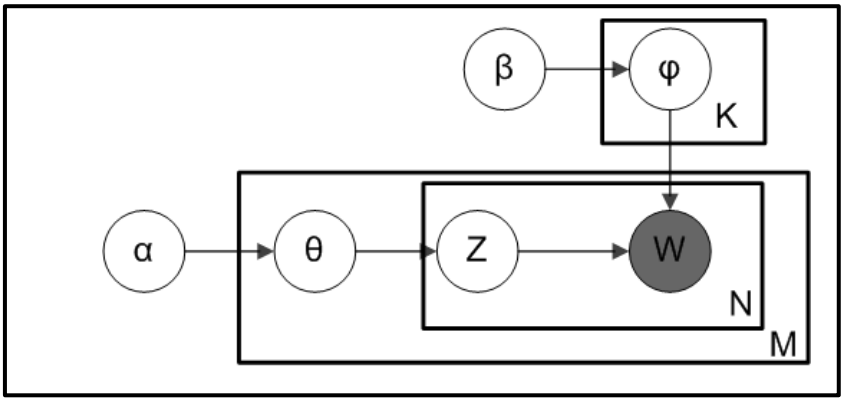


Figure 2-4 Graphical model representation of LDA [55]

2.9.2 LDA2Vec

Christopher Moody presented LDA2Vec in [15], a hybrid approach that combines the sparse document-topic representation of a corpus utilized in traditional topic modeling techniques such as the LDA with the innovation of word embeddings. LDA2Vec was developed for solving one of the fundamental limitations of traditional topic modeling approaches: the lack to benefit from recent advances in distributed word representation, such as the Skip-gram representation presented in Word2vec. As previously discussed, word embeddings have had a significant impact on the field of NLP across a wide range of applications, and the development of hybrid topic models that leverage these distributed word representations to account for semantically meaningful relationships between words has led to advances in topic modeling as well as the development of state-of-the-art topic models.

From an implementation standpoint, the LDA2Vec model modifies the Skip-gram Negative Sampling (SGNS) objective from [47] to learn document-wide feature vectors in synchronization with discovering document weights that are loaded onto topic vectors [15]. Figure 2.5 illustrates the LDA2Vec model workflow and details of the model are provided below.

- **Loss Function**

The total loss term L in (4) is the sum of the Skip-gram Negative Sampling loss L_{ij}^{neg} with the addition of a Dirichlet likelihood term over document weights, L^d which is discussed later. The loss is conducted using a context vector,c_j, pivot word vector w_j, target word vector w_i, and negatively-sampled word vector w_l [20].

$$L = L^d + \Sigma_{ij} L_{ij}^{neg} \quad (4)$$

$$L_{ij}^{neg} = \log \sigma\,(\vec{c_j} \cdot \vec{w_i}) + \Sigma_{i=0}^{n} \log \sigma\,(-\vec{c_j} \cdot \vec{w_i}) \quad (5)$$

- **Word Representation**

The Skip-gram Negative Sampling loss indicated in (5) tries to distinguish context word pairs that exist in the corpus from those randomly sampled from a "negative" pool of words. This loss is reduced when the observed words are totally segregated from the marginal distribution. Pairs of pivot and target words (*j*,*i*) are recovered when they co-occur in a moving window scanning through the corpus. The pivot word is used to anticipate the neighboring target word for every pivot target pair of words. Each word is represented as a fixed-length dense distributed representation vector, which is utilized in both the pivot and target representations [20].

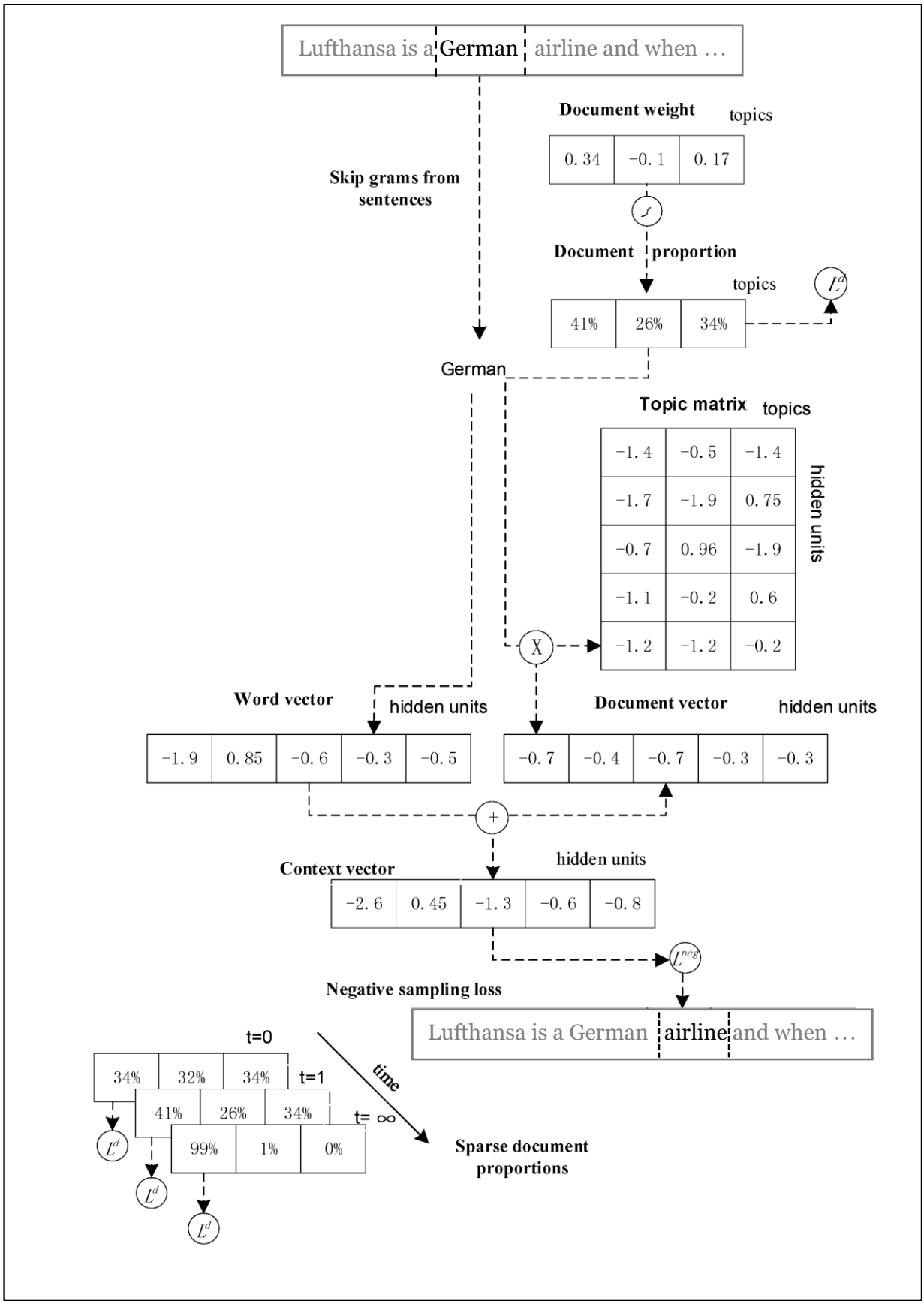


Figure 2-5 The LDA2Vec model's workflow [15]

- **Document Representation**

LDA2Vec embeds words and document vectors in the same space and trains both representations at the same time. Both spaces are effectively connected by combining the pivot and document vectors. The context vector in LDA2Vec is intentionally meant to be the sum of a document vector and a word vector, as seen in (6) [15]:

$$\overrightarrow{c_j} = \vec{\omega}_j + \vec{d}_j \qquad (6)$$

- **Document Mixture**

LDA2Vec generates a document vector from a mixture of topic vectors by constraining the document vector d_j to project onto a set of latent topic vectors $t_0, t_1, \ldots, t_k$. Each weight is a fraction that denotes the membership of document *j* in the topic *k* [20].

$$\vec{d}_j = p_{j0} \cdot \vec{t}_0 + p_{j1} \cdot \vec{t}_1 + \cdots + p_{jk} \cdot \vec{t}_k \qquad (7)$$

- **Sparse Membership**

The document weights p_{ij} are sparsified by optimizing the document weights with respect to a Dirichlet likelihood with a low concentration parameter α:

$$L^d = \lambda \Sigma_{jk} (\alpha - 1) \log p_{jk} \qquad (8)$$

The overall objective in (8) assesses the likelihood of document j in topic k summed across all available documents. The intensity of this term is regulated by the tuning parameter lambda. This simple likelihood promotes the document proportions coupling in each topic to be sparse when alpha is less than (1) and homogenous when alpha is more than (1) [20].

2.10 Clustering validation methods

Clustering is an unsupervised learning approach in which labels are not given or, in certain situations, do not exist. Although there are automated and semi-automatic data labeling approaches, they may not be accurate enough to confirm clustering.

Cluster validation is a technique for measuring the efficiency of various clustering algorithms. This aids in avoiding identifying patterns in a random dataset. There exist three types of cluster validation approaches to evaluate the goodness of the clustering result [57].

- **Internal cluster validation:** evaluates the goodness of a clustering structure using internal information from the clustering process without referring to external information (having no labels).
- **External cluster validation:** comparing cluster analysis results to an externally known outcome, such as class labels. This approach is typically used to select the optimal clustering algorithm for a given data set.
- **Relative cluster validation:** examines the clustering structure by modifying the parameters for the same technique (For example, varying the number of clusters k). It is frequently used to determine the optimal number of clusters.

Because data labels exist in this study, external measures are used, which consist of seven measures: Purity, Precision, Recall, F1-score, Rand-Index (Accuracy), Jaccard Index, and Normalized Mutual Information. They are some of the most often used validation methods in the literature, and they are employed in this study to validate the quality of the clustering. The sub-sub sections that follow will provide a very brief overview of the mentioned evaluation metrics [57].

Some of these measures are calculated using the widely used computing confusion matrix, which is made up of four values: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) [58]. As seen in Table 2.2, these components combine to generate the Confusion Matrix.

Table 0-2 Confusion Matrix

	Same Cluster	Different Cluster
Similar documents	True Positives (TP)	False Negatives (FN)
Different documents	False Positives (FP)	True Negatives (TN)

The clustering confusion matrix is built on all possible combination-pairs of all documents selected from all clusters [7], where:

- **TP**: denotes that the two documents are similar and belong to the same cluster.
- **FN**: denotes that the two documents are similar and belong to different clusters.
- **FP**: denotes that the two documents are different and belong to the same cluster.
- **TN**: denotes that the two documents are different and belong to different clusters.

1. Purity

Purity is a metric for determining how pure a cluster is in relation to its dominant class [7]. The percentage of all objects of dominating classes in each cluster in relation to the total number of objects is then used to calculate purity:

$$Purity = \frac{1}{N}\sum_{k} max_j \left| w_k \cap c_j \right| \qquad (9)$$

Where N is the total number of objects, k denotes the number of clusters, and w_k the dominating class, whereas the true class is c_j . The higher the purity value, the better the clustering, with a maximum value of one of the dominant classes in a cluster representing all objects in that cluster.

2. Precision

Precision (P) is the ratio of the number of relevant documents to the total number of documents in all clusters [59].

$$P(i,j) = \frac{n_{i,j}}{n_j} \qquad (10)$$

Where $n_{i,j}$ is the number of correct members of the class label i in cluster j, and n_j is the total number of members of the cluster number j.

Based on the confusion matrix values, precision can be calculated according to the following formula:

$$Precision = \frac{TP}{TP + FP} \qquad (11)$$

3. Recall

For each cluster, the Recall metric is dependent on the relevant class label. The recall is calculated by dividing the total number of relevant documents in the collection by the number of relevant documents in all clusters [60].

$$R(i,j) = \frac{n_{i,j}}{n_i} \qquad (12)$$

Where $n_{i,j}$ is the number of correct members of the class i in cluster j, an n_i is the number of original members of the class number i.

Based on the confusion matrix values, recall can be calculate according to the following formula:

$$Recall = \frac{TP}{TP + FN} \qquad (13)$$

4. F1-score

F1-score [61] is a common external quality metric that combines information retrieval precision and recall ideas. It's also known as the F-Score or the F-Measure. The F1-score of cluster j and class i is calculated as follows:

$$F(i,j) = \frac{2 \times P(i,j) \times R(i,j)}{P(i,j) + R\ (i,j)} \qquad (14)$$

5. Rand Index (Accuracy)

The Rand Index calculates the percentage of accurate clustering decisions. Clustering may be thought of as a sequence of pair-wise selections in which it is intended to assign two documents to the same cluster if and only if they are similar [62]. The formula for calculating the Rand index is as follows:

$$Rand - index = \frac{TP + TN}{TP + TN + FP + FN} \qquad (15)$$

6. Jaccard Index

The Jaccard index measures the similarity between the clustered values and the true values to assess the capability of the clustering technique to find similarities between data objects and performance based on these similarities [63]. According to the confusion matrix, The Jaccard index is defined as follows:

$$Jaccard - index = \frac{TP}{TP + FN + FP} \quad (16)$$

7. Normalized Mutual Information (NMI)

Normalized Mutual Information is an information-theoretic measure of how much information is shared between a clustering and a ground-truth classification that may discover nonlinear similarity between cluster members [63]. The NMI is calculated using (17).

$$NMI(X, Y) = \frac{I(X, Y)}{\sqrt{H(X) \times H(I)}} \quad (17)$$

Where: X represents class labels and Y represents cluster labels. I(X, Y) represents mutual information between X and Y. H(X) and H(Y) indicates the entropy of X and Y, respectively. A higher NMI value indicates more mutual information, resulting in, more similarity between clusters

CHAPTER THREE

THE PROPOSED APPROACH

3.1 Introduction

In this chapter, the steps followed to achieve the key aim of the current thesis are described. This includes proposing an approach for clustering Arabic documents using the topic modeling technique. The architecture of this model is depicted first; then, the stages of text pre-processing, text representation, training model, and testing model are discussed subsequently.

3.2 Outline of the Proposed Approach

There are five stages in the proposed approach. Each stage comprises sub-steps in order to accomplish the study objectives and meet its main goal. As described in Figure 3.1, these steps are Text collection, Text pre-processing, Text representation, Topic Modeling model, and Evaluation.

The first stage is collecting text data; the dataset utilized in this approach comprises a collection of documents, the contents and sources of which will be identified later. Text pre-processing is the second one, which includes a number of sub-steps to prepare the proposed approach inputs. The third one is the text representation; after constructing the vocabulary taken from the dataset, documents represents as document vectors and the word embeddings vectors will be represented in one of two ways: learning embeddings from scratch or using Pre-trained word vectors. The topic modeling model using LDA2Vec algorithm is described in the fourth stage. The final stage is the evaluation of the outcomes of the model, and many methods were used to do so.

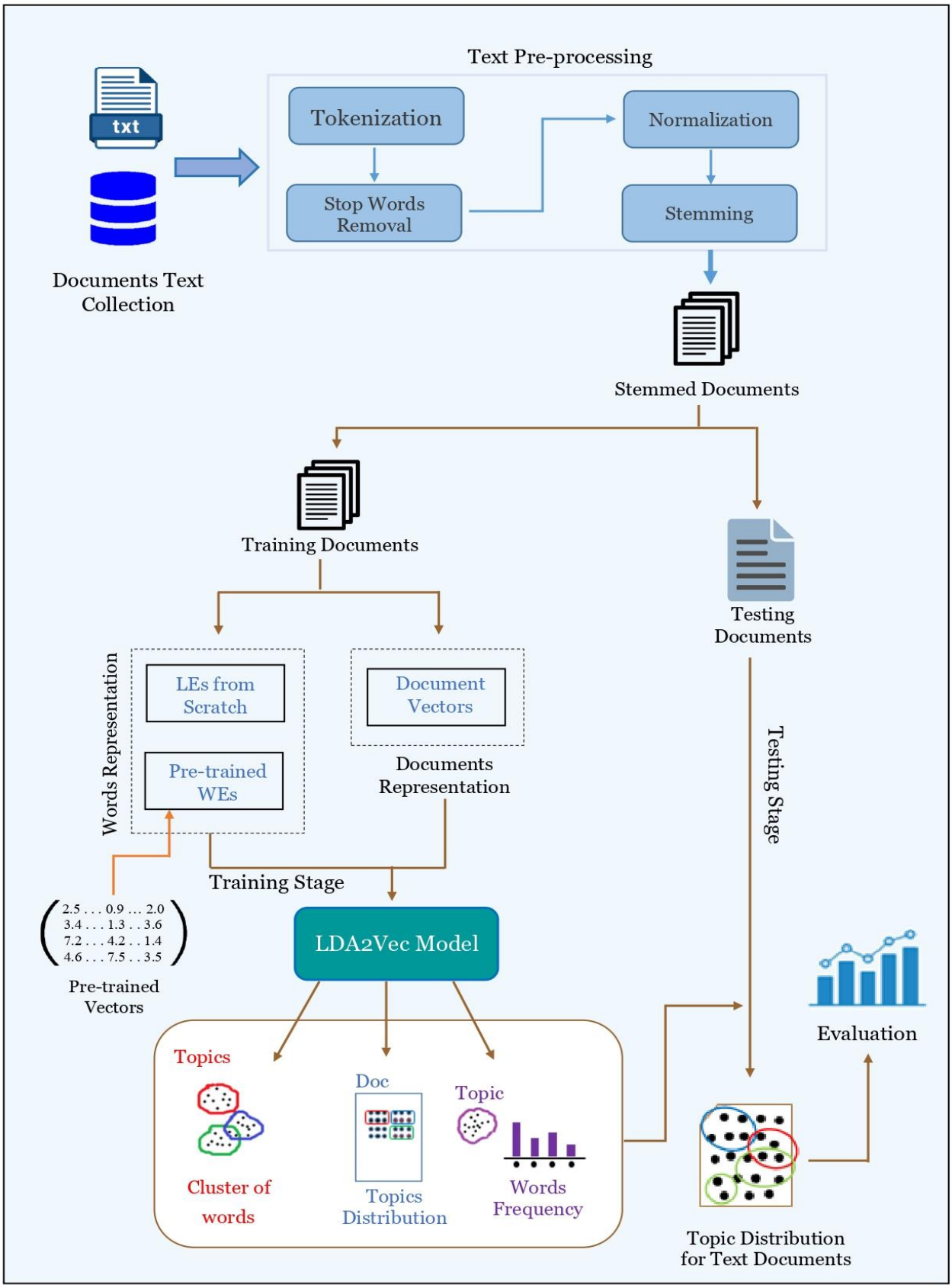


Figure 3-1 The proposed Approach Diagram

3.3 Collecting Text Data

In the current approach, Arabic articles news dataset collected by Diab Abuaiadah et.al, [64] has been used. Where the researchers chose the number, type, and source of categories as the initial stage in creating the dataset. After examining numerous published articles on Arabic document classification and scanning from well-known and trustworthy Arabic websites, nine primary categories were chosen: Art, Economy, Law, Politics, Literature, Health, Religion, Technology, and Sport. Documents from each discipline were manually collected and sorted, with each document weighing up at roughly 2 kilobits. Each document is allocated to one category, so every document that looks to belong to more than one was removed from the dataset. Table 3.1 lists the sources for each category.

Table 3-1 The Dataset

Category	NO Docs.	Size (MB)	Sources
Art	300	1.01	http://www.egypty.com/all-arts.aspx; http://www.elcinema.com/news/articles/2010/12/
Economy	300	1.01	http://news-all.com/; http://www.spa.gov.sa/;http://all4syria.info http://www.aljazeera.net/ebusiness/
Health	300	1.01	http://www.se77ah.com; http://www.aljazeera.net; http://www.6abib.com/
LAW	300	1.02	http://www.eastlaws.com/News/NewsList.aspx; http://www.barasy.com/
Literature	300	0.98	http://www.almhml.com/ ; http://news-all.com/ ; http://adab.akhbarway.com
Politics	300	1.00	http://news-all.com/ ; http://www.spa.gov.sa/ http://all4syria.info; http://www.aljazeera.net
Religion	300	1.08	http://news-all.com/ ; http://www.anbacom.com/
Sport	300	1.01	http://www.kooora.com/ ; http://www.soccerarabia.net/
Technology	300	1.06	http://news-all.com/; http://www.akhbarway.com/; http://www6.mashy.com

There are five versions of the dataset:

1) **V1**: Documents with no preprocessing.
2) **V2**: Documents with stop words removed.
3) **V3**: Documents after stop words removed and stemmed with the Light10 algorithm.
4) **V4**: Documents after stop words were removed and stemmed with Chen's algorithm.
5) **V5**: Documents after stop words removed and stemmed with Khoja's algorithm.

They can be accessed through http://diab.edublogs.org/dataset-for-arabic-document-classification/, version 1 is used, which contains the raw data. There are 2,700 documents, 300 in each category, with 878,726 terms and 96,859 unique terms [7].

3.4 Text Pre-processing

The second stage of the proposed approach is pre-processing. Text documents are preprocessed at this stage since the raw data is not in the optimal format. Text documents are cleaned of unimportant and undesirable data at this stage. In the current thesis, several pre-processing methods are used, as indicated in the following subsections. Algorithm 3.1 depicts the pre-processing steps of text documents that have raw data.

3.4.1 Tokenization

Tokenization is carried out in this approach by breaking out the text document into words depending on the spaces between them. The tokenization method for each text document is explained in Step 2 of Algorithm 3.1.

3.4.2 Removing Stop words

The procedure of removing stop words[1] is shown in Step 3 of Algorithm 3.1. A large list of Arabic stop words was used in this approach. It contains 750 stop words. The researcher added additional stop words, increasing the total to 872. Table 3.1 shows examples of Arabic stop words.

Table 3-2 Examples of Arabic Stop Words

Arabic word	Arabic word	Arabic word	Arabic word	Arabic word	Arabic word
إن	أنا	لك	لكن	ماذا	كلا
كم	فيها	كيفما	على	إذا	نعم
أنتَ	يوم	كان	من	الى	اللذان
الذي	ذلك	لِمن	في	ليس	لستُ
إليك	الآن	الأول	هو	هي	هم
اما	مع	متى	أحيانا	ثم	منذ
بعد	عند	الف	بين	الذي	ايضا
بعض	عندما	غدا	اي	به	ثلاثة
اعادة	اجل	انه	دون	امام	الذاتي
بسبب	عليه	اول	حول	ايام	الاخيرة

3.4.3 Text Cleaning and Normalization

It is not possible to move directly from raw text to fitting a machine learning or deep learning model; first, the text must be cleaned. There are several text preparation methods that may be necessary, and the method of choice is totally based on the natural language processing tasks. The process of cleaning text in this approach includes, removing punctuation,

[1] https://github.com/mohataher/arabic-stop-words

special characters, numbers, non-Arabic letters, Arabic diacritics, normalizing the Arabic Alef alphabet, and Ta-Marbuta.

1. Removing Punctuations and Special characters

In-text pre-processing, removing unnecessary information such as punctuations and special characters is essential. Punctuations explain how a sentence is constructed, how it should be read, and how it should be understood. Examples of punctuations and special characters are: '!', '"', '#', '$', '%', '&', '"', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '؟', '@', '[', '\\', ']', '^', '_', '`', '{', '|', '}', '~' . The procedures for removing punctuation and special characters from text are explained in Step 4.1 of Algorithm 3.1.

2. Removing Numbers

Because the model deals with text, the numbers may not provide much information for text processing, so they are removed from the text. Step 4.2 in Algorithm 3.1 explains the procedure followed in removing numbers from texts.

3. Removing Non-Arabic Letters

Algorithm 3.1, Step 4.3, describes the process of removing non-Arabic letters from texts. All uppercase [A-Z] letters from A to Z, as well as all lowercase [a-z] letters from a to z, are included.

4. Removing Arabic diacritics

Arabic diacritics (short vowels) are diacritical marks put above or below ordinary consonant letters. These diacritics are the fatha الفَتْحَة, the kasra الكَسْرَة, and the dahmma الضَّمَّة. In addition, there are two other essential marks: the sukoon السُّكُون and the shaddah الشَّدَّة. Step 4.4 of Algorithm 3.1 explains removing Arabic diacritics procedure.

5. Normalizing Arabic Alef alphabet

Alef or (Alif) is the most utilized letter in Arabic owing to the several sounds it represents. In addition, this leads to numerous different forms and shapes to distinguish each sound. The different forms of the alef [ٱإآا] have been replaced with plain alef (ا) as explained in step 4.5 in Algorithm 3.1.

6. Normalizing Ta- Marbuta

Step 4.6 in Algorithm 3.1 illustrates normalizing the feminine ending, Ta-Marbuta ة, to ha ه.

Algorithm 3.1: Pre-processing of Text Documents

Input: Texts is a set of text documents

Output: Set of tokens (SOT)

Begin:

1. Step 1: // Read Texts
2. Step 2: // Splitting text into tokens (Tokenization)
3. For T in Texts do
4. Extract all tokens (ET) based on space between words
5. Tokens_set = ET // Set of tokens
6. End For
7. Step 3: // Removing Stop words
8. i = 1
9. LSW = list of Arabic stop words
10. While i <= length of Tokens_set then
11. **Begin**

```
12.       If Token [i] existed in LSW then
13.          replace Token with white space
14.          i = i + 1
15.       End If
16.    End while
17. Step 4:    // Removing unimportant tokens and normalization
18.    i = 1
19.    While i <= length of Tokens_set do
20.    Begin
21.       If Token [i] in Tokens_set then
22.          Step 4.1:   RP = Removing punctuations and special characters
                         from Tokens_set
23.          Step 4.2:   RN = Removing numbers from Tokens_set
24.          Step 4.3:   RNAL = Removing non-Arabic letters from Tokens_set
25.          Step 4.4:   RAD = Remove Arabic diacritics from Tokens_set
26.          Step 4.5:   RAlef = Replacing  ٱ ,إ ,أ, آ, by ا in Tokens_set
27.          Step 4.6:   RTa = Replacing ة by ه in Tokens_set
28.                      Replace RP, RN, RNAL, RAD with
                         white space
29.          i = i +1
30.       End If
31.    End while
32. Return set of tokens that contains important tokens (SOT)
End
```

3.4.4 Arabic Stemming

Words in the Arabic language are made up of root sets, which show the basic meaning of a word with the addition of affixes that change the pronunciation. Accordingly, Arabic Light Stemmer based on root extraction stemmer (Information Science Research Institute (ISRI)) is used in this study. The utilized stemmer comprises of numerous phases as indicated in Algorithm 3.2.

Algorithm 3.2: Arabic Light Stemmer

Input: Set of Tokens (SOT)

Output: LS, list of light stemming words

Begin:

```
1.   SOT, list of words from preprocessing step
2.   LS={ }
3.   For w in SOT do
4.       If w not in LSW then      //exclude stop words from being
                                     Processed
5.           w = removing ("و")    //remove connective 'و'
6.           w = removing prefix32 //remove prefixes with lengths three and two
7.           w = removing suffix32  // remove suffix with length three and two
8.           If  w.Length >= 3     // keep words with length 3 or greater
9.               LS.append(w)
10.          End If
11.      End If
12.  End For
13.  Return list of stemmed words LS
End
```

3.4.5 Calculating word frequency

Word frequency investigates the significance of words in a text or group of texts by counting the number of times particular words appear. Raw and relative frequency counts, as well as percentages, are included. Algorithm 3.3 explain the steps of counts the frequency of each word in the text.

Algorithm 3.3: Word frequency

Input: ***LS***, List of stemmed words

Output: dictionary of words and its counts

Begin:

```
1.  counts = ( )
2.  For word in LS:
3.      If word in counts:
4.          counts[word] += 1
5.      Else:
6.          counts[word] = 1
7.  End For
8.  Return counts
```

End

3.4.6 Removing Low-Frequency Words

One of the difficulties in dealing with text is that the number of feature words accessible in the dataset is just too huge. The number of features available might possibly number in the tens of thousands. Limiting the number of input features is recommended in order to decrease modeling

computational costs and, in some cases, improve model performance. In current study, all terms that contain less than (25) occurrences are eliminated.

3.4.7 Creating Vocabularies

First, a list of unique words from the text documents was compiled. Such words are referred to as vocabulary. A list of features is produced from all of the words, with each feature consisting of a pair (key, token). The key indicates the feature's index in the list, while the token refers to the feature itself. The vocabulary of the standard dataset has 96,859 features. In the current approach, the python package (vocab) has been used to create the vocabulary as illustrated in Algorithm 3.4.

Algorithm 3.4: Creating Vocabularies

```
from vocab import Vocab
import numpy as np
```

Input: **documents:** (the input documents)

max_length: (the maximum number of words per document, If the document is shorter than this number, it will be padded to this length)

skip: (int) Short documents will be padded with this variable up until max_length)

nlp: is a spaCy NLP object. Useful for not instantiating the object multiple times.

Output: vocab, dictionary (Keys are the word index, and values are the string. The pad index gets mapped to None)

Begin:

```
1. voc = Vocab()
2. If nlp is None:
3.     nlp = Arabic()
4.     data = np.zeros((len(documents), max_length) , dtype='int32')
```

```
5. data[:] = skip
6. For row, doc in enumerate(nlp.pipe(documents)):
7.       text = str(doc)
8.       data[row ,:length] = dat[:length]
9.       dat = voc.word2index(s.split(), train=True)
10. uniques = np.unique(data)
11. vocab = {v: voc.index2word(v) for v in uniques if v != skip}
12. vocab[skip] = '<SKIP>'
13. return vocab
End
```

The following example explains the result of building vocabularies process:

Document 1
" كشفت شركة مايكروماكس الهندية للهواتف المحمولة عن دخولها سوق الإمارات العربية المتحدة "
Document 2
" تساهم شركة هارمان هاوس مع شركة سامسونج على تعزيز حضورها في الإمارات العربية المتحدة "
vocab = {0: 'كشفت', 1: 'شركة' , 2: 'مايكروماكس', 3: 'الهندية', 4: 'للهواتف', 5: 'المحمولة', 6: 'عن', 7: 'دخولها', 8: 'سوق', 9: 'الإمارات' , 10: 'العربية', 11: 'المتحدة', 12: 'تساهم', 13: 'هارمان', 14: 'هاوس', 15: 'مع', 16: 'سامسونج', 17: 'على', 18: 'تعزيز', 19: 'حضورها', 20: 'في'}

3.5 Creating a Vector of Features

One of the most fundamental problems in text mining and information retrieval (IR) is text representation. The text was represented as a feature vector space, with each document being an array of features. The length of the vector is the same for all documents because it depends on the length

of the feature. To prevent having features that are overly huge, tokens are only considered features if they occur more frequently than other features.

3.5.1 Documents Representation

The documents in this model are represented by BOW model. BOW turns text into a matrix of word occurrences within a document. The collection of documents is represented as a Term Document Matrix (TDM), which is a m x n matrix with the following properties:

- Rows (i=1,m) indicate words from the collection of documents.
- Columns (j=1,n) represent documents from the documents collection.
- Cell ij stores the occurrences of the word i in the context of the document j.

Table 3.3 illustrate documents representation in Term Document Matrix.

Table 3-3 Term-Document matrix

	D1	D2	D3	D4
W1	1	4	3	1
W2	3	0	1	2
W3	0	5	4	6
W4	0	1	0	2
W5	6	3	0	0

3.5.2 Words Representation (Word Embeddings)

1. Learning Embeddings from Scratch

In this case, the word embedding vectors are created from scratch. The Word2vec algorithm was used to generate word embeddings vectors

from a document collection in which words with related meanings are represented similarly.

2. Pre-trained Word Embeddings

Pre-trained word embeddings are embeddings that have been learned in one task and may be utilized to solve another comparable task. As they are taught on huge datasets, Pre-trained word embeddings capture a word's semantic and syntactic meaning. They have the ability to improve an NLP model's performance.

3.6 Topic Modeling

Once the pre-processing stage has completed its role, the pre-processed text will be sent to the topic-modeling system. The topic-modeling system will attempt to find 'topics' from the pre-processed text.

The LDA2Vec has been used as a topic-modeling technique in this approach to determine and group the various topics covered by 2700 documents. The system contains two stages, the training stage, and the testing stage.

3.6.1 LDA2Vec Model /Training Stage

LDA2Vec is developed by modifying the Skip-gram Negative-Sampling (SGNS) aims to use document-wide feature vectors while concurrently learning continuous document weights loading onto topic vectors. In LDA2Vec, the power of Word2vec is paired with the interpretability of LDA. This recipe requires three architectural changes:

1. Bringing together global document topics and local word patterns.
2. Word vectors that are dense but document vectors that are sparse.
3. Mixture models for interpretability.

The procedure of the LDA2Vec technique is described in section 2.9.2.

In the training stage, The LDA2Vec method has some parameters. The parameters include the number of documents (n_docs), number of topics (n_topics), number of vocabulary (n_vocab), number of dimensions in a single word vector (n_units), negative sampling power (power), number of negative samples (n_samples), term frequency (term_frequency) and sampling temperature (temperature).

The pre-processed text has been passed through the LDA2Vec method to extract hidden/latent topics and saves the extracted topics as feature topics. Algorithm 3.5 explain the LDA2Vec Method procedure.

Algorithm 3.5: LDA2Vec Method

Input: n_docs, n_topics, n_units, n_vocab, term_frequency, n_samples, power, temperature, doc_lengths, pretrained,

Output: Loss data for Negative Sampling

Begin:

1. **LDA2Vec** (n_docs, n_topics, n_units, n_vocab, term_frequency, n_samples, power, temperature)
2. Step1: (A latent vector is randomly initialized for every document in the corpus)
   - Given an array of document integer indices, returns a vector for each document. The vector is composed of topic weights projected onto topic vectors. According to the equation:

$$\vec{d}_j = p_{j0} \cdot \vec{t}_0 + p_{j1} \cdot \vec{t}_1 + \cdots + p_{jk} \cdot \vec{t}_k$$

3. Step 2: Calculate the log likelihood of the observed topic proportions. According to the equation:

$$L^d = \lambda \Sigma_{jk} (\alpha - 1) \log p_{jk}$$

4. Step 3: Create context vector by sum the document vector and word vector. According to the equation:

$$\overrightarrow{c_j} = \overrightarrow{\omega_j} + \overrightarrow{d_j}$$

5. Step 4: Using the negative sampling to calculate the gradient for a few sampled negative examples. According to the objective function explained in the equations:

$$L = L^d + \Sigma_{ij} L_{ij}^{neg}$$

$$L_{ij}^{neg} = \log \sigma (\overrightarrow{c_j} \cdot \overrightarrow{w_i}) + \Sigma_{i=0}^{n} \log \sigma (-\overrightarrow{c_j} \cdot \overrightarrow{w_i})$$

End

For model successfully trains and identifies a set of topics for a given dataset, the output saved in an npz file (a zipped archive of files called after the variables they contain) that includes topic_term_dists vectors, doc_topic_dists vectors, doc_lengths, vocab, and term_frequency values. Algorithm 3.6 describes the LDA2Vec training stage procedure.

Algorithm 3.6: LDA2Vec Model-Training Stage

Input: n_docs, n_topics, n_units, n_vocab, term_frequency, n_samples, power, temperature, doc_lengths, pretrained, epochs

Output: npz file (data), which containes (topic_term_dists vectors, doc_topic_dists vectors, doc_lengths, vocab, and term_frequency values).

Begin:

1. Step1: Train_model = LDA2Vec (n_docs, n_topics, n_units, n_vocab, term_frequency, n_samples, power, temperature)
2. Step2: **If** pretrained = True:
   // use pre-trained vectors
3. Step3: **For** epoch in range (epochs)
4. **Begin**:

5. Step3.1: data = Collects a dictionary of word, document and topic distributions.
6. Step3.2: data['doc_lengths'] = doc_lengths
7. Step3.3: data['term_frequency'] = term_frequency
8. Step3.4: Predict word given context and pivot word.
9. **End For**
10. **Return** data
11. **End**

3.6.2 LDA2Vec Model /Testing Stage

As a test stage, a set of documents is picked as a test set from the dataset. Then, one by one, these documents are labeled with certain topics. Using the model to identify the topics of test documents, started by examining each word in the test document to see whether it corresponded to the training stage vocabulary. The probability of these words is calculated using the topic_term_dists vectors that obtained from training stage. Then, based on the total of the word probabilities, analyzed the summation of the word probabilities for each document, and the topic with the highest probability is the topic of this document. Algorithm 3.7 illustrate the procedure for this stage.

Algorithm 3.7: LDA2Vec Model-Testing Stage

Input: stemmed_words is a pre-processed test document,
n_topics is the number of topics,
data is the output of the training stage.

Output: doc_to_topic is a topics distribution of test documents

Begin:

1. Step1 Initialized doc_to_topic = [[]]*len(stemmed_words)

```
2. Step2:    Initialized variable  m=0
3. Step3:    For text in stemmed_words:
4.           Begin:
5.               Step3.1: tokens = str(text).split()
6.               Step3.2: topic_weight = np.zeros(n_topics)
7.               Step3.3: For token in tokens:
8.                        Begin:
9.                            If token in data['vocab']:
10.                               ind = data['vocab'].index(token)
11.                               For i in range(n_topics):
12.                                   topic_weight[i] += data['topic_term_dists'][i][ind]
                                                         /len(tokens)
13.                           End For
14.                       End If
15.                   End For
14.              Step3.4: doc_to_topic[m] = topic_weight
15.              End For
16. Return doc_to_topic
17. End
```

3.7 Evaluation

Since, the 'topics' generated by the LDA2Vec model can be considered as clusters, so can applied different clustering evaluation methods to test clustering accuracy and quality, including: Purity, Precision, Recall, F1-score, Normalized Mutual Information, Accuracy or Rand-Index, Jaccard Index (explained in section 2.10), and testing topic distribution per document.

CHAPTER FOUR

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

This chapter discusses the results of each stage of the proposed approach, which was described in chapter three. The results of all stages are organized in the order in which they appeared in Chapter Three. This chapter also compares the outcomes of the proposed approach to the results of comparable studies. On the other hand, the current chapter starts with the hardware and software requirements for implementing the proposed approach.

4.2 Hardware and Software Requirements

The proposed approach is implemented using the following hardware and software requirements.

Hardware: Processor Intel i7, RAM 8GB, Storage 1000 GB, Freq.1.8GHz 2.00GHz**,**

Software: Operating System: Windows10 pro-64-bit.

Programming language: Python language

IDLE: the approach is implemented by Python 3.8.3 Jupyter Notebook, Anaconda 3.

4.3 The Dataset

The dataset that has been used to train the model is presented by [64] as described in section (3.3) of chapter three. The dataset has five versions; we utilized Version 1, which includes the original documents. Figure 4.1 shows an example of a document and the document's content. The dataset comprises 2,700 documents organized into nine categories, each with 300 documents. The total word count is 878,726, with a vocabulary size of 96,859 and an average length of 1966.68. Figure 4.2 displays a histogram of documents lengths.

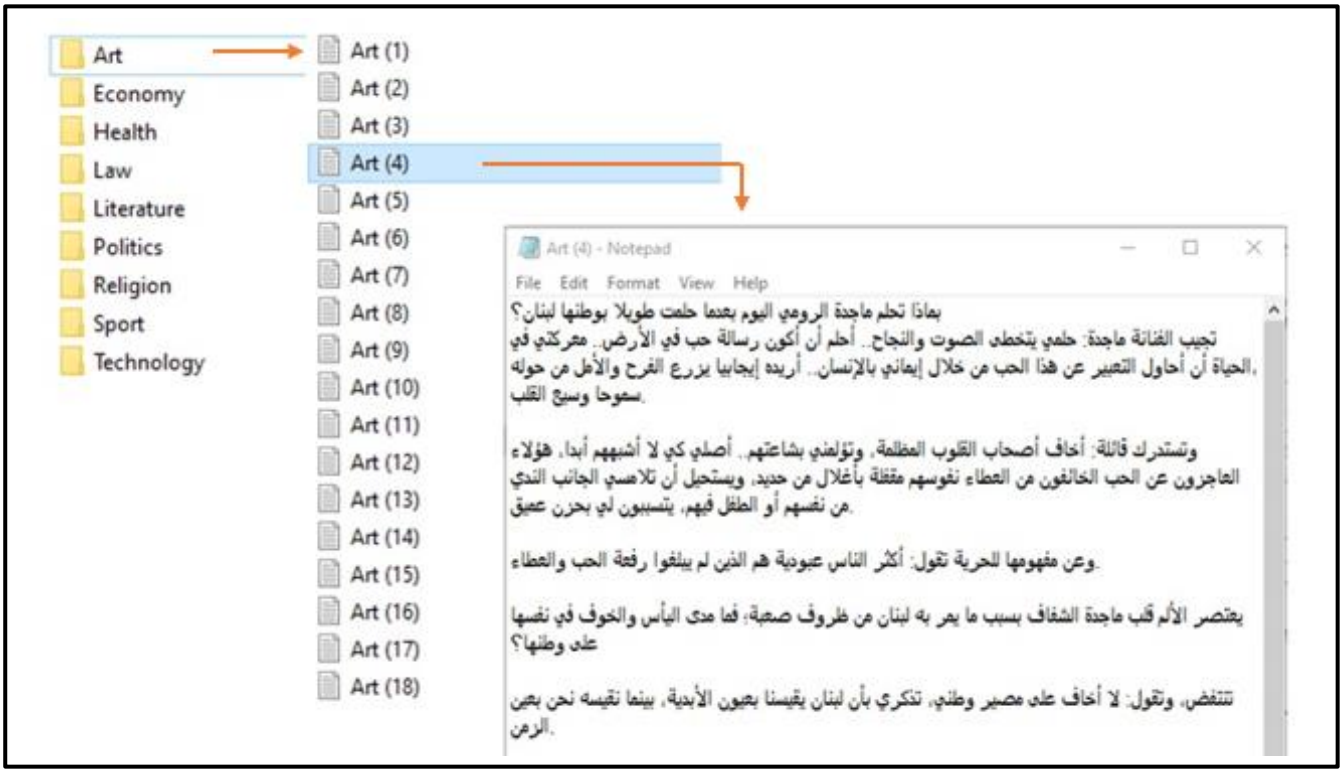


Figure 4-1 Sample of a text document

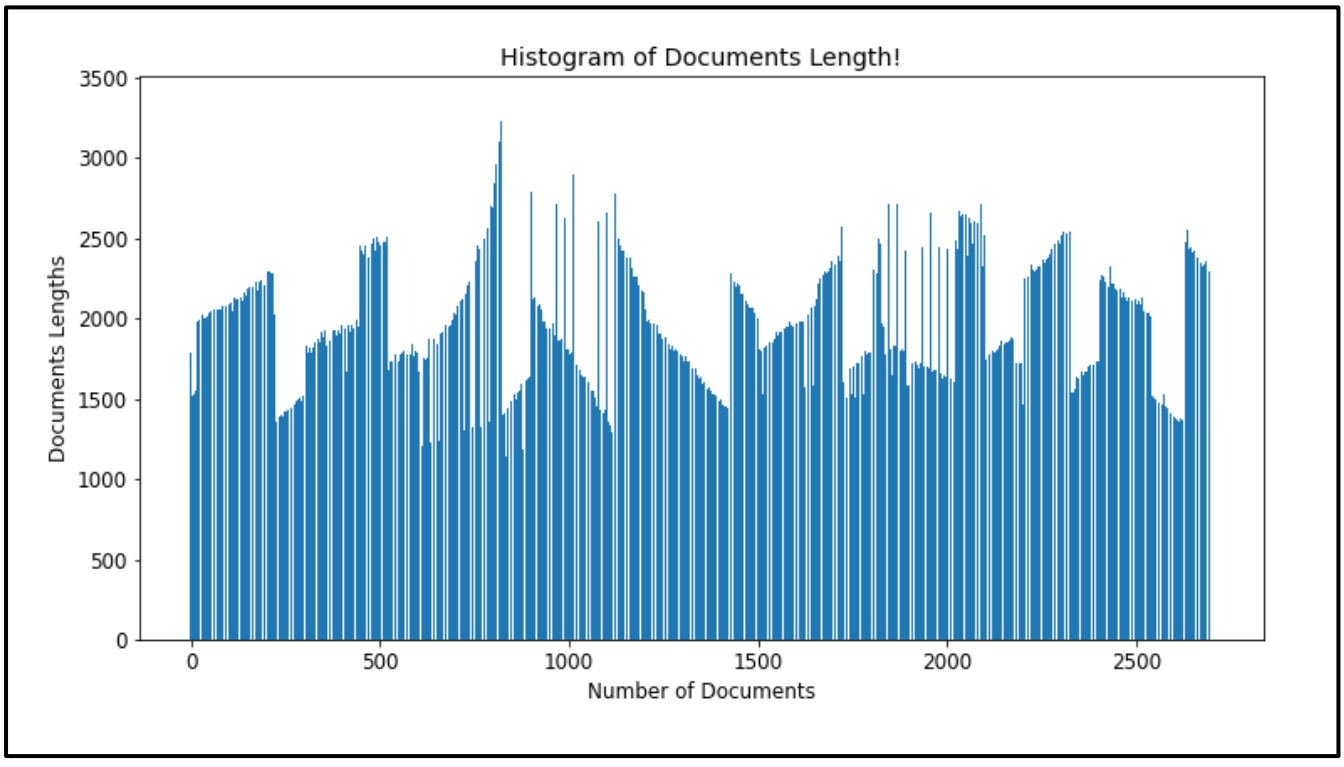


Figure 4-2 Histogram of documents lengths

4.4 Results of Text Pre-processing

The results of the pre-processing stage are shown after performing tokenization to split words, cleaning text, and normalization by removing irrelevant information. The outcomes of the pre-processing stage were a collection of tokens.

- **Tokenization**

Table 4.1 displays the results of splitting the text of a document into its constituent words.

Table 4-1 Tokenization

Text before Tokenization	لم يكن يتم عمل إختبار لفيروس (سي) في الدم المأخوذ من المتبرعين قبل عام 1993 فى مصر (1992 فى البلاد الأخرى).
Text after Tokenization	['لم', 'يَكن', 'يتم', 'عمل', 'إختبار', 'لفيروس', '(', 'سي', ')', 'فى', 'الدم', 'المأخوذ', 'من', 'المتبرعين', 'قبل', 'عام', '1993', 'فى', 'مصر', '(', '1992', 'في', 'البلاد', 'الأخرى', ')', '.']

- **Removing Arabic Stop Words**

In-text mining, unimportant words, known as stop words, are removed from the original text. Table 4.2 demonstrates a sample of text before and after removing stop words.

Table 4-2 Removing Arabic Stop words

Text before Tokenization	يذكر أن الفنان عزت العلايلى قد أعلن اعتذاره عن عدم المشاركة في المسلسل بعد قراءته مجموعة الحلقات الأولى، وقال إنه لم يجد نفسه في دور تاجر المخدرات «سلطان».
Text after Tokenization	يذكر الفنان عزت العلايلى أعلن اعتذاره المشاركة المسلسل قراءته مجموعة الحلقات الأولى، يجد دور تاجر المخدرات « سلطان » .

- **Removing Punctuations and Special Characters**

Punctuation and special characters have been removed from all texts. Table 4.2 shows an example of the text before and after the punctuation and special characters are removed.

Table 4-3 Removing Punctuations and Special Characters

Text before Removing Punctuations and Special Characters	وكان البنك المركزي الأوروبي قد ضخ الجمعة 61 مليار يورو، (قرابة 84 مليار دولار) في القطاع المصرفي، على شكل أوراق مالية.
Text after Removing Punctuations and Special Characters	وكان البنك المركزي الأوروبي قد ضخ الجمعة 61 مليار يورو قرابة 84 مليار دولار في القطاع المصرفي على شكل أوراق مالية

- **Removing Numbers**

The numbers are removed because they may not provide much information for text processing. Table 4.4 shows a sample of the original text before and after the numbers are removed.

Table 4-4 Removing Numbers

Text before Removing Numbers	وكشف المحمودي أنه يوجد في الإمارات حوالي 700, 10 سيدة أعمال منهن 4095 في دبي و 3000 في الشارقة 732, 2 في أبوظبي.
Text after Removing Numbers	وكشف المحمودي أنه يوجد في الإمارات حوالي , سيدة أعمال منهن في دبي و في الشارقة , في أبوظبي.

- **Removing Non-Arabic Letters**

Table 4.5 demonstrates the result of removing non-Arabic letters from the text.

Table 4-5 Removing Non-Arabic Letters

Text before Removing Non-Arabic Letters	أعلنت الشركة العالمية "إبسون" (Epson)، عن إطلاقها جهاز نسخ الأقراص الليزرية وأقراص "دي. في. دي" (CD/DVD) الجديد "ديسكبروديوسر بي. بي-100" (Discproducer PP-100) في أسواق الشرق الأوسط.
Text after Removing Non-Arabic Letters	أعلنت الشركة العالمية "إبسون" ()، عن إطلاقها جهاز نسخ الأقراص الليزرية وأقراص "دي. في. دي" (/) الجديد "ديسكبروديوسر بي. بي-100" (-100) في أسواق الشرق الأوسط.

7. Removing Arabic diacritics

The Arabic language features short vowels that produce varied pronunciations. They are grammatically required but are removed in written Arabic texts. The result of removing Arabic diacritics from the text is shown in Table 4.6.

Table 4-6 Removing Arabic diacritics

Text before Removing Arabic diacritics	من الأسباب المعنوية الدعاء، فالله قريب مجيب: {وَإِذَا سَأَلَكَ عِبَادِي عَنِّي فَإِنِّي قَرِيبٌ أُجِيبُ دَعْوَةَ الدَّاعِ إِذَا دَعَانِ فَلْيَسْتَجِيبُوا لِي وَلْيُؤْمِنُوا بِي لَعَلَّهُمْ يَرْشُدُونَ}
Text after Removing Arabic diacritics	من الأسباب المعنوية الدعاء، فالله قريب مجيب: {وإذا سألك عبادي عني فإني قريب أجيب دعوة الداع إذا دعان فليستجيبوا لي وليؤمنوا بي لعلهم يرشدون}

8. Normalizing theArabic Alef alphabet

Replacing the different forms of the alef [أإآا] with plain alef (ا) as seen in the example in Table 4.7

Table 4-7 Normalizing Arabic Alef alphabet

Text before Normalizing Alef	استعادت أسعار النفط عافيتها خلال التعاملات الأسيوية الأربعاء عقب انخفاضها إلى ما دون 70 دولار للبرميل متأثرة بضعف قوة الإعصار "دين" وتراجع المخاوف من تأثيره السلبي على إنتاج النفط المكسيكي
Text after Normalizing Alef	استعادت اسعار النفط عافيتها خلال التعاملات الاسيوية الاربعاء عقب انخفاضها الى ما دون 70 دولار للبرميل متاثرة بضعف قوة الاعصار "دين" وتراجع المخاوف من تاثيره السلبي على انتاج النفط المكسيكي

9. Normalizing Ta- Marbuta

Table 4.8 shows an example of how to normalize the feminine ending, Ta-Marbuta ة, to ha ه

Table 4-8 Normalizing Ta- Marbuta

Text before Normalizing Ta- Marbuta	أصبحت الأدوية المتاحة في الصيدليات بدون وصفة طبية موضوعا نقاشيا بين أهل الاختصاص من حيث فوائدها وآثارها الجانبية وتفاعلاتها الضارة.
Text after Normalizing Ta- Marbuta	أصبحت الأدويه المتاحه في الصيدليات بدون وصفه طبيه موضوعا نقاشيا بين أهل الاختصاص من حيث فوائدها وآثارها الجانبيه وتفاعلاتها الضاره.

- **Arabic Stemming**

This involves removing affixes like prefixes and suffixes from words. This can assist in the reduction of the number of words in the feature space. Table 4.9 provides an illustration of text after Arabic light stemming.

Table 4-9 Arabic Stemming

Text before using Arabic Stemming	اعتمدت اليابان على الطرق التقليدية في مواجهة الأزمة فلجأت إلى سلاح سهل هو تخفيض معدلات الفائدة لتأمين سيولة كافية ورخيصة وتخفيض سعر الصرف الحقيقي.
Text after using Arabic Stemming	اعتمدت ياب طرق تقليدية مواجهة ازمة فلجأت سلاح سهل تخفيض معدل فائدة لتأم سيولة كافية رخيصة تخفيض سعر صرف حقيقي.

10. Creating Vocabularies

The set of unique tokens in the documents collection is referred to as the vocabulary. From all of the words, a list of features is created, with each feature consisting of a pair of words (key, token). Figure 4.3 presents some of the vocabulary found in the dataset, which had 49,592 vocabularies after pre-processing.

11. Removing Low-Frequency Words

Limiting the amount of input features is recommended in order to decrease modeling computational costs and, in some cases, improve model performance. All words with less than **twenty-five** occurrences are removed from vocabulary. Figure 4.4 shows the top 25 most frequent words in the dataset. The horizontal axis represents the word repetitions, while the vertical axis refers to the most repeated words.

{0: 'بررت',1: 'ساندي', 2: 'اتهامات', 3: 'لقرار', 4: 'بسحب', 5: 'اغنيه', 6: 'استوديو', 7: 'اثار', 8: 'هاني', 9: 'محروس', 10: 'تسحب', 11: 'لديه', 12: 'بوضع', 13: 'برغم', 14: 'تاكيد', 15: 'مادي', 16: 'ادبي', 17: 'انها', 18: 'ستتعا', 19: 'البوم', 20: 'هانى', 21: 'وقرر', 22: 'معاقبه', 23: 'فعلته', 24: 'بطرح', 25: 'مواقع', 26: 'انترنت', 27: 'عربيه', 28: 'لبنانيه', 29: 'خساره', 30: 'مبالغ', 31: 'كبيره', 32: 'يتولى', 33: 'انتاج', 34: 'بومات', 35: 'بنفس', 36: 'امام', 37: 'لجوء', 38: 'قضاء', 39: 'تحصل', 40: 'حقها', 41: 'فقامت', 42: 'بعمل', 43: 'محضر', 44: 'يعمل', 45: 'تتهمه', 46: 'بتسريب', 47: 'قامت', 48: 'تعرض', 49: 'ارسل', 50: 'رساله', 51: 'تهديد', 52: 'تليفو', 53: 'محمول', 54: 'هاتفه', 55: 'شخصي', 56: 'جدير', 57: 'ساندى', 58: 'تنوى', 59: 'اسواق', 60: 'يتضمن', 61: 'اغني', 62: 'سيطرح', 63: 'توزيع', 64: 'احمد', 65: 'ابراهيم', 66: 'امير', 67: 'طعيمه', 68: 'محمد', 69: 'رفاعى', 70: 'توقفت', 71: 'اعمال', 72: 'فنيه', 73: 'قاهره', 74: 'يعني', 75: 'تاجيل', 76: 'نوال', 77: 'جديد', ...}

Figure 4-3 Some of the vocabularies found in the dataset

['الله', 'رئيس', 'شركه', 'عالم', 'فريق', 'عربيه', 'نقطه', 'جديده', 'محمد', 'مجلس', 'جديد', 'مجموعه', 'مستوى', 'فيلم', 'مائه', 'مباراه', 'دوله', 'منتخب', 'مؤشر', 'كتاب', 'اتحاد', 'بطوله', 'نظام', 'احمد', 'كبير']

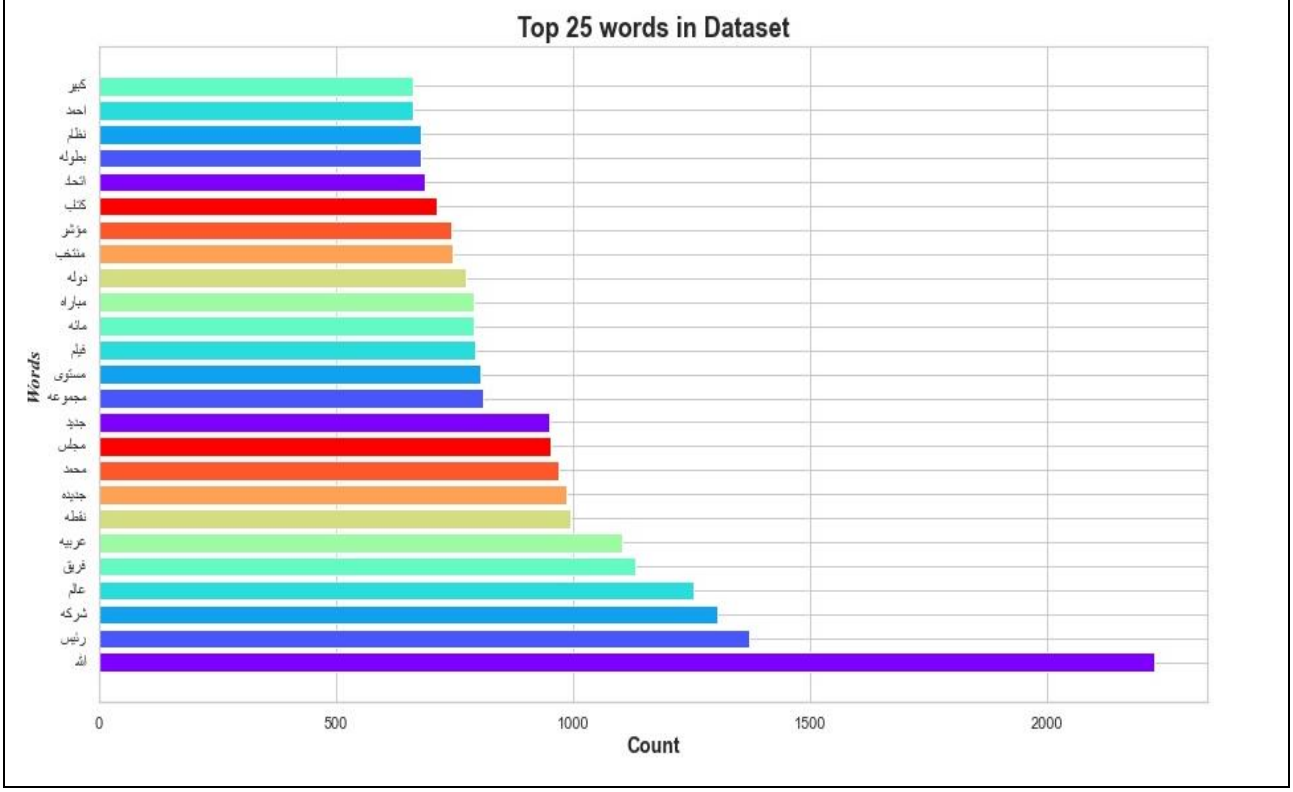


Figure 4-4 Top 25 most frequent words in the dataset.

Following the completion of the pre-processing procedures, the text was completely devoid of unnecessary information that may have influenced the proposed approach's accuracy. Using a Word Cloud, Figure 4.5 visualizes text before preprocessing while Figure 4.6 visualizes getting text after preprocessing.

Figure 4-5 Text before implementing the Pre-processing.

Figure 4-6 Text after implementing the Pre-processing

4.5 Results of Words Representation

1. Learning Embeddings from scratch

In this case, the word embedding vectors are constructed from scratch using the words2vec skip-gram model outlined in section 2.8.2. The vectors of word embeddings are generated from a document collection in which words with comparable meanings are represented similarly.

2. Pre-trained Word Embeddings

- fastText (cc.ar.300.vec): 300-dimensional vectors trained on Wikipedia using a fastText [65] is used as pre-trained word embeddings in the current study. The pre-trained vectors for a specific vocab are visualized in Figure 4.7.

```
1 vectors['شركه']
```

```
array([-0.0199,  0.0126,  0.063 ,  0.0248,  0.1511, -0.0593, -0.0758,
        0.0444, -0.023 , -0.0467, -0.1482, -0.0416,  0.1118,  0.1142,
       -0.1207,  0.0422,  0.0795,  0.1134, -0.1506,  0.0303,  0.0572,
       -0.1076,  0.0236,  0.1018, -0.0396,  0.0078, -0.0172,  0.0207,
        0.1064,  0.0127, -0.0552, -0.008 ,  0.129 , -0.023 , -0.0709,
        0.1218,  0.1664, -0.0398,  0.0019, -0.0665, -0.0741, -0.1068,
       -0.0705,  0.0617,  0.0865,  0.0544, -0.0899, -0.006 , -0.044 ,
        0.0422, -0.0078,  0.0678, -0.0486, -0.0475, -0.1071,  0.0108,
        0.0511, -0.173 ,  0.0083,  0.0355,  0.1349,  0.0083, -0.0209,
        0.0392,  0.0004,  0.0364, -0.1587,  0.0809, -0.0367,  0.0748,
        0.009 , -0.036 ,  0.0337,  0.0842, -0.0907,  0.0782, -0.072 ,
       -0.1107, -0.0579,  0.0762, -0.0664, -0.0493,  0.0502,  0.0342,
       -0.0963,  0.0636, -0.0206,  0.0596,  0.0118,  0.0572, -0.1456,
        0.0771, -0.1052,  0.1098, -0.1029,  0.0864, -0.0609,  0.0767,
       -0.052 ,  0.0464, -0.0796, -0.0931,  0.0688,  0.006 , -0.0072,
        0.0076,  0.0819,  0.0953, -0.0237,  0.0108,  0.0494,  0.0276,
        0.045 , -0.1278, -0.0069,  0.0272,  0.064 ,  0.0731, -0.0378,
       -0.0714,  0.0953, -0.0267,  0.0775,  0.0415, -0.0388, -0.0038,
        ...
```

Figure 4-7 Pre-trained vectors for a specific vocab

4.6 Results of the LDA2Vec Model

The data set comprises 2,700 documents divided into a training set and a test set with split ratios of 70-30 and 80-20, respectively.

Following the pre-processing stage and stemming, the dataset is contained:

1. **In 70-30 ratio:** 350,629 total words with a vocabulary size of 41,295 in 1890 documents.
2. **In 80-20 ratio:** 400,718 total words with a vocabulary size of 44,171 in 2160 documents.

During the training stage, the LDA2Vec technique is used to cluster and extract "topics/clusters" from the dataset. The configuration of the experiments follows the methodology of the proposed approach specified in Figure 3.1. The LDA2Vec model is trained across the dataset in individual mini-batches (128) at a time using the Adam optimizer for two hundred epochs and the number of topics is (9).

In the testing stage, the number of documents is (810 and 540) in 30-70 and 80-20 ratios, respectively. New documents testing is carried out in accordance with the procedure stated in section 3.6.2.

To analyze the accuracy and quality of generated topics/clusters, metrics such as purity, precision, recall, F1-score, accuracy (rand index), Jaccard-index, and normalized mutual information are applied, as well as examining the topic distribution over documents.

- **Experiments and Results**

Several experiments are conducted to evaluate the validity of the proposed approach in clustering Arabic text documents, as shown below.

- The **first set of experiments** is carried out to evaluate a range of LDA2Vec parameters are evaluated by varying the negative sampling exponent (power) β = 0.5; 0.75; 1.0 and the number of "negative" samples k = 5; 15, covering the two techniques of words representation, learning embeddings from scratch and pre-trained word embeddings. Tables 4.10 and 4.11 show the results of the first set of experiments.

Table 4-10 Comparison of LDA2Vec model results with negative sampling power β = 0.5; 0.75; 1.0, Negative Sampling number k=5 and split ratio 70-30

Dataset Split	Training 70% Testing 30%					
Negative sampling Power	β = 0.5		β = 0.75		β = 1.0	
Negative Sampling number	K=5		K=5		K=5	
Word Representation	LEs from scratch	Pre-trained WEs	LEs from scratch	Pre-trained WEs	LEs from scratch	Pre-trained Wes
Purity	0.8358	0.8604	**0.8913**	<u>0.8641</u>	0.8259	0.8259
Precision	0.8698	0.8926	**0.9019**	<u>0.8753</u>	0.8404	0.8476
Recall	0.7802	0.8271	**0.8654**	<u>0.8617</u>	0.7925	0.7777
F1-score	0.8155	0.8471	**0.8793**	<u>0.8658</u>	0.8109	0.7994
Rand index (Accuracy)	0.9070	0.9214	<u>0.9362</u>	**0.9397**	0.9168	0.9144
Jaccard index	0.6396	0.7052	**0.7627**	<u>0.7570</u>	0.6564	0.6363
NMI	0.6991	0.7533	**0.7784**	<u>0.7723</u>	0.7068	0.7211

Bold numbers indicate the best value while <u>underlined</u> numbers represent the second-best value.

Table 4-11 Comparison of LDA2Vec model results with negative sampling power β = 0.5; 0.75; 1.0, Negative Sampling number k=5 and split ratio 80-20

Dataset Split	Training 80% Testing 20%					
Negative sampling Power	β = 0.5		β = 0.75		β = 1.0	
Negative Sampling number	K=5		K=5		K=5	
Word Representation	LEs from scratch	Pre-trained WEs	LEs from scratch	Pre-trained WEs	LEs from scratch	Pre-trained WEs
Purity	0.7685	0.8018	**0.8888**	0.8833	0.8314	0.8574
Precision	0.7674	0.7517	0.8951	**0.8987**	0.9091	0.8936
Recall	0.7407	0.7666	0.8515	**0.8833**	0.7185	0.8092
F1-score	0.7379	0.7461	0.8712	**0.8826**	0.7890	0.8341
Rand index (Accuracy)	0.9121	0.9195	0.9520	**0.9551**	0.8920	0.9369
Jaccard index	0.5882	0.6216	0.7419	**0.7910**	0.5606	0.6796
NMI	0.6979	0.7390	0.8026	**0.8334**	0.6598	0.7923

As the best results are achieved with the 80-20 split ratio and negative sampling power β = 0.75, the model is trained by changing the number of negative samples K=15 and comparing the results to those obtained with K= 5, while keeping negative sampling power = 0.75 constant. Table 4.12 displays the results obtained.

Table 4-12 Comparison of the LDA2Vec model results with a number of negative samples k = 5, 15 and negative sampling power β = 0.75.

Data split	Training 80% Testing 20%			
Negative sampling Power	β = 0.75		β = 0.75	
Negative Sampling number	K=5		K=15	
Word Representation	LEs from scratch	Pre-trained WEs	LEs from scratch	Pre-trained WEs
Purity	**0.8888**	0.8833	0.8074	0.7759
Precision	0.8951	**0.8987**	0.8506	0.8505
Recall	0.8515	**0.8833**	0.7018	0.7611
F1-score	0.8712	**0.8826**	0.7527	0.7690
Rand index (Accuracy)	0.9520	**0.9551**	0.8687	0.8919
Jaccard index	0.7419	**0.7910**	0.5406	0.6143
NMI	0.8026	**0.8334**	0.6472	0.7482

In brief, the model produces better results when negative sampling power β = 0.75, the number of "negative" samples k = 5, and the highest accuracy is achieved when using pre-trained word embeddings for word representation.

- As the results show, increasing the number of training documents has improved the accuracy of the model.
- By considering the "topics" generated in earlier experiments, the best topics discovered are achieved with the 80-20 split ratio, β = 0.75, k = 5 when utilizing pre-trained word embeddings for word representation, the model discovered nine topics as described in Table 4.13.

Table 4-13 Topics discovered by LDA2Vec model.

Topic ID	Topic Label	Top Words
Topic 1	Law	جرائم , اختصاص , قضائيه , قضاء , دستور , قانونيه , محكمه , مكتب , متهم , سلطه
Topic 2	Religion	الله , مسلم , حجاب , تعالى , احمد , كريم , دعاء , انظر , يهود, حديث
Topic 3	Sport	مباراه , منتخب , بطوله , ملاعب , بايرن , استراليا , فريق , بطول , موسم , يلعب
Topic 4	Economy	اسهم , اسمنت , سعري , بلغت , سعريه , خسائر , صفقه , مؤشر , قيمه , تداول
Topic 5	Politics	سياسيه , حكومه , دوليه , سوريا , اسرائيل , امريكي , عداله , اوسط , اسرائيليه , بلغت
Topic 6	Literature	كتاب , روايه , مكتبه , شعريه , قصائد , قصيده , ادبي , شاعر , احتفاليه , محور
Topic 7	Health	عاده , طبيعيه , دماغ , التهاب , مزمن , اطفال , الدم , معده , اثناء , اسباب
Topic 8	Art	غناء , جمهور , اغنيه , شاشه , مطرب , تامر , تصوير , نجوم , كليب , جديد
Topic 9	Technology	حاسوب , مكتب , شاشه , شبكه , جهاز , رقميه , قيمه , عملاقه , شركه , اتصال

1. The **second set of experiments** involves a verification experiment using LDA, which is one of the most common topic modeling techniques. The LDA model is trained using the same data as the LDA2Vec training, which produced the best results. The generated topics are then evaluated using the previously mentioned evaluation techniques; by comparing these results to the LDA2Vec results, it can be noticed that the LDA2Vec outperforms the LDA, as stated in Table 4.14 and part of these results are displayed in Figure 4.8.

Table 4-14 Comparison of LDA and LD2Vec results

Model	LDA	LDA2Vec
Purity	0.7537	**0.8833**
Precision	0.8644	**0.8987**
Recall	0.7537	**0.8833**
F1-score	0.7644	**0.8826**
Rand index (Accuracy)	0.8929	**0.9551**
Jaccard index	0.6047	**0.7910**
NMI	0.7144	**0.8334**

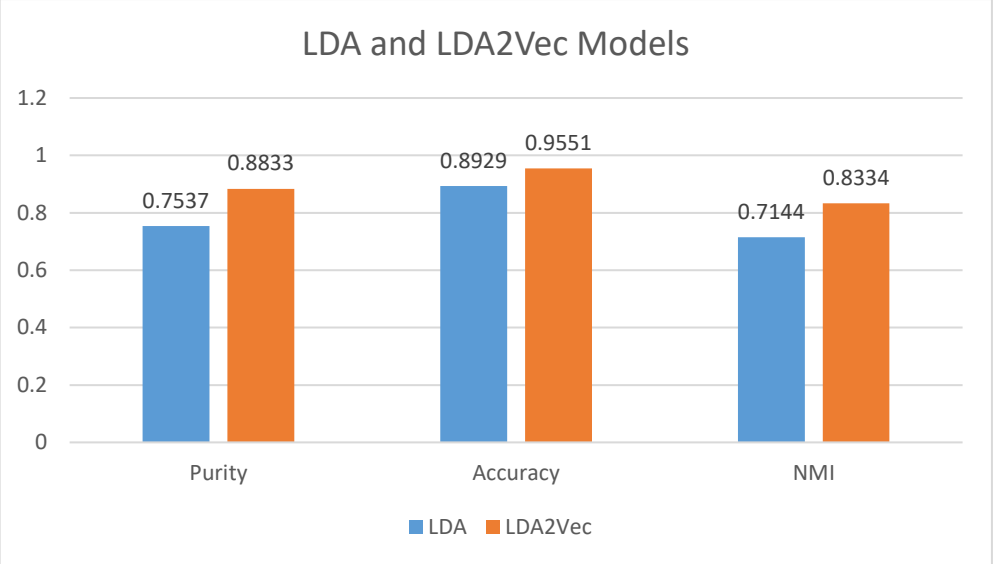
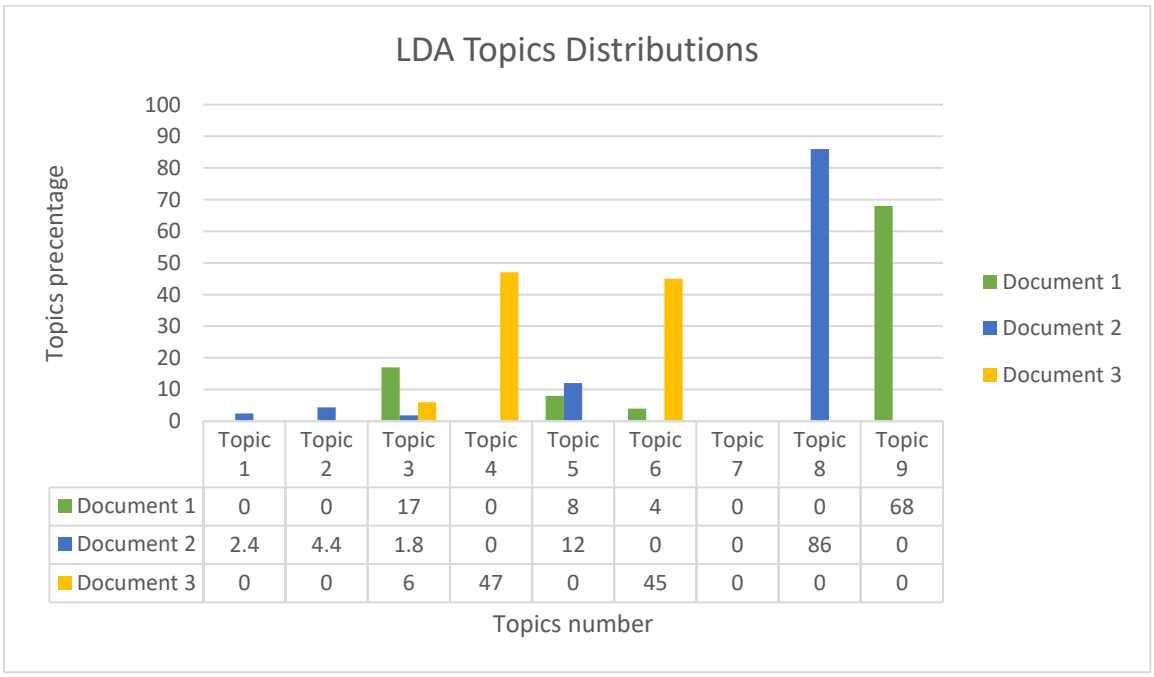


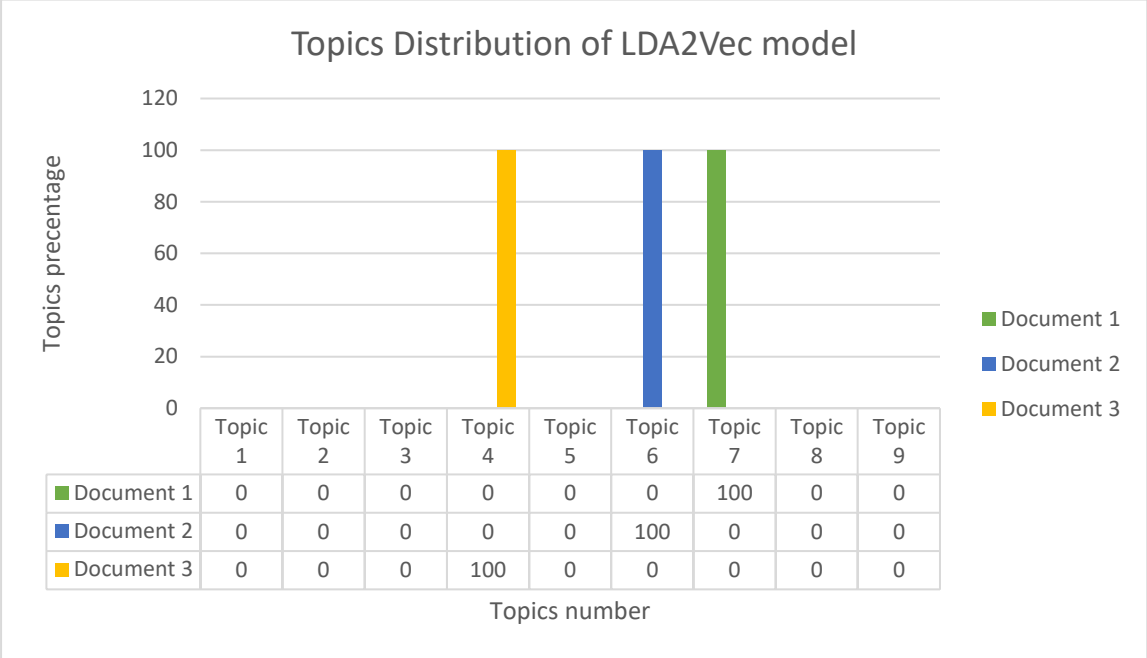
Figure 4-8 Comparison of LDA and LD2Vec results

As each document in the dataset is assigned to one category, and after implementing the models, the distribution of topics has been examined in three random documents from the training set, and the results are given in Figures 4.9 and 4.10:



	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
Document 1	0	0	17	0	8	4	0	0	68
Document 2	2.4	4.4	1.8	0	12	0	0	86	0
Document 3	0	0	6	47	0	45	0	0	0

Figure 4-9 Three random documents' topic distribution after implementing LDA model.



	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
Document 1	0	0	0	0	0	0	100	0	0
Document 2	0	0	0	0	0	100	0	0	0
Document 3	0	0	0	100	0	0	0	0	0

Figure 4-10 Three random documents' topic distribution after implementing the LDA2Vec model.

To summarize, The LDA is not particularly dependable, as it can be seen from the results. The topics created by the model for documents are not always accurate, which is the main reason for the model's decreasing accuracy. Otherwise, LDA2Vec uses Word2vec to generate a document vector, which improves its accuracy in the context of (word-topic), probabilistic distribution. As it is noted in Figure 4.10, the model has a significantly better distribution of topics. e.g., document (1) assigned to 100% of topic 7, while it belongs to more than one topic in the LDA model and in different proportions as shown in Figure 4.8.

2. In the **third set of experiments**, the best results of the proposed approach in this study are compared with the results of a method used in a similar study for the same purpose on the same dataset, where authors utilized a combined method (LDA and K-means algorithm) to cluster Arabic documents. Table 4.15 demonstrate that the proposed approach in this study performed much better than the method in [7].

The strength of LDA2Vec comes in the fact that it not only learns word embeddings (and context vector embeddings) for words, but it also learns topic and document representations at the same time.

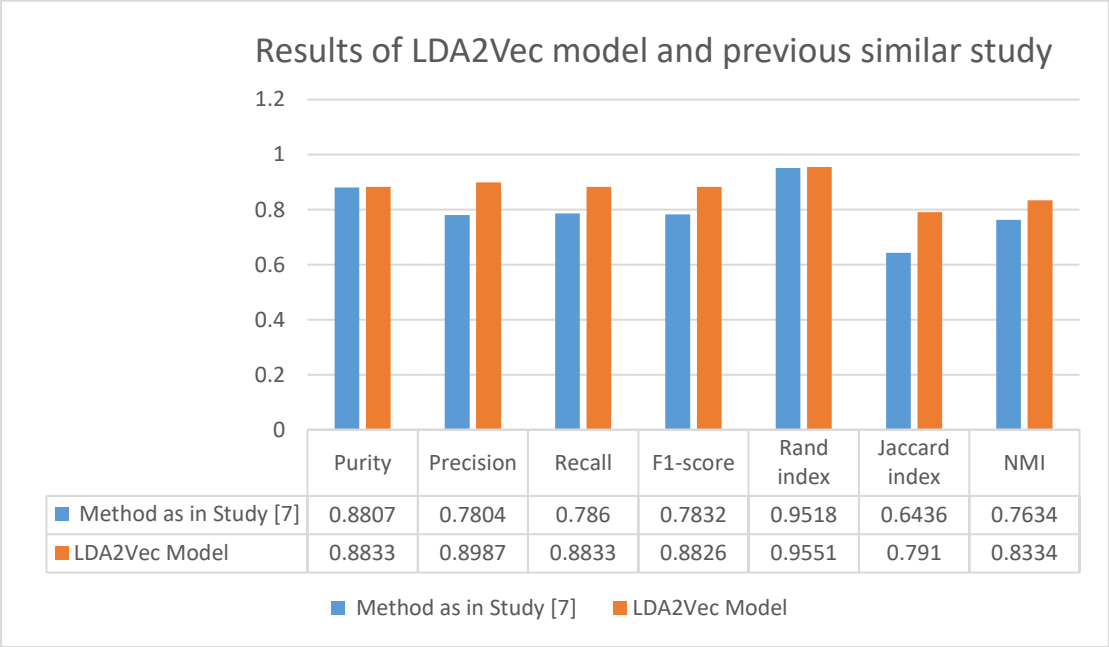


Figure 4-11 Comparison of LDA2Vec Model Results to previos similar study [7]

4.7 Visualization of LDA2Vec Model Results

1. By utilizing a Seaborn Heatmap (which is a graphical representation of data where values are depicted by color). It highlights the relationship between documents and topics (doc_topic_distribution). As shown in Figure 4.11, the document names represent the Y-axis, the topic labels represent the X-axis, and the color represents the accuracy of the document's distribution in this topic, with the darker the color representing a higher percentage of the topic distribution in this document.

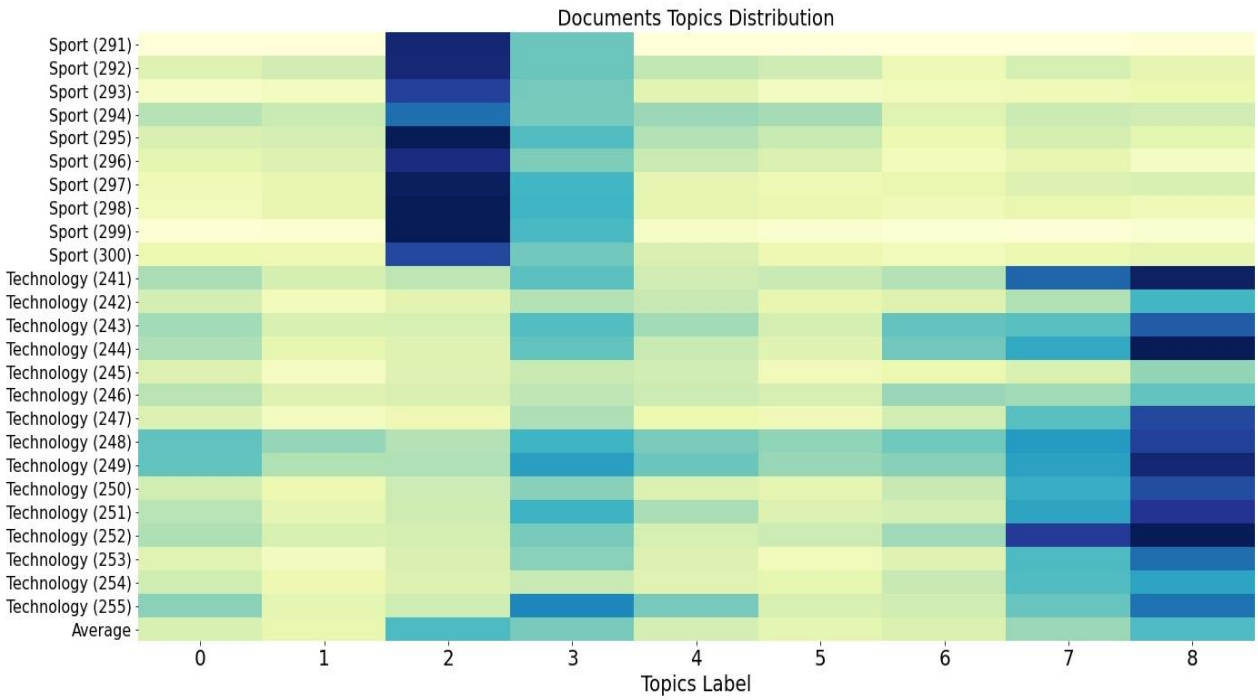


Figure 4-12 Documents Topics Distribution

2. Took a snapshot of the pyLDAvis output as shown in Figure 4.12. The area of the circle shows the relevance of each topic over the whole dataset, while the distance between the centers of the circles indicates topic similarity. The histogram on the right side listed the top 30 most relevant terms for each topic. LDA2Vec assisted in extracting (9) main topics (as shown in Figure 4.13). For example, in topic two, the most relevant terms found are ، اختصاص ، جرائم دستور ، قضاء ، قضائيه, etc. and this is very likely a topic for Law.

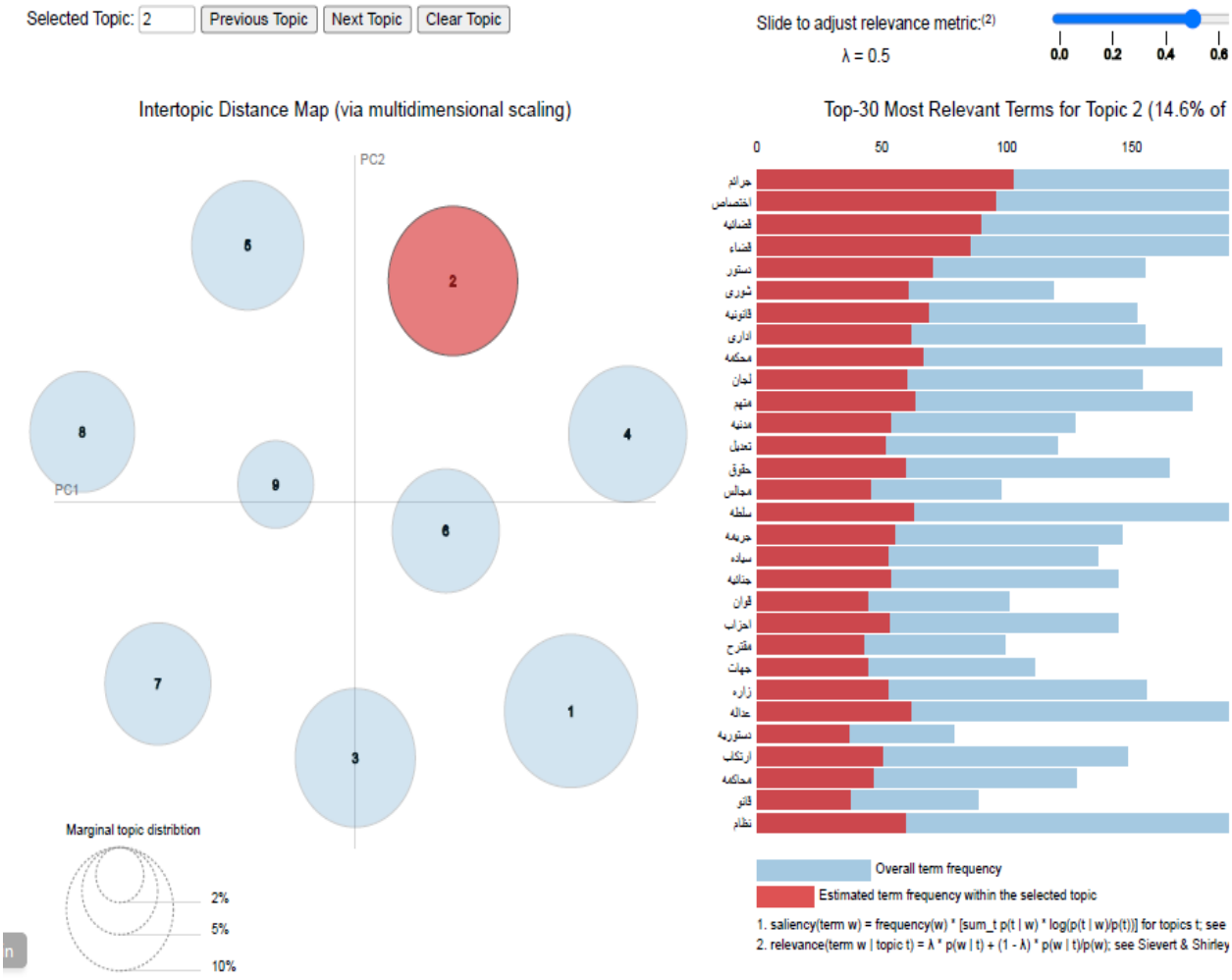


Figure 4-13 LDA2Vec Topics Visualization

4.8 Case Study

In practice, the proposed approach has been used to cluster theses for postgraduate students at Babylon University in the Central Library. It was used by collecting the titles and abstracts of theses for various fields. Table 4.15 illustrated the details of the collected data.

Table 4-15 Collected Data Details

File No.	Thesis Title	Researchers Names	Year	Thesis Field
1	الكشف الجزيئي والتقييم المناعي لبعض مسببات الأمراض البكتيرية في التهابات الجهاز التنفسي	نور نبيل عباس هدى هادي محمد الحسناوي	2018	Health
2	الاضطرابات في تخطيط الدماغ الكهربائي الكمي في الأطفال المصابين بمرض طيف التوحد وعلاقتها بالخلل الوظيفي في اللغة والسلوك	شمس كريم عبد حمادي زاهد محمد علي كاظم	2022	Health
3	السؤال بوصفه من وسائل الرقابة البرلمانية على اعمال مجلس الوزراء - دراسة مقارنة	محمد فاهم سلمان عدنان عاجل عبيد	2017	Law
4	تحليل المشاعر باللغة العربية لتحديد مؤيدي الإرهاب على تويتر باستخدام تقنيات تنقيب عن البيانا ت	عصام كاظم محمد احمد العزاوي	2019	Information Technology
5	كشف أنتهاك أتفاقية مستوى الخدمة في مجال عمليات العمل من خلال تعليم نظام التصنيف	حوراء عبد الأمير صبح احمد خلفة العجيلي	2022	Information Technology

The title and abstract of each thesis were saved in a separate file, and these files were uploaded to the proposed system, and the distribution of topics for each file was tested. The highest probability topic was picked as a topic for this file (thesis), and the results are summarized in Table 4.16.

Table 4-16 Results of Case Study

File No.	Topic ID	Topic Label
1	7	Health
2	7	Health
3	1	Law
4	9	Technology
5	9	Technology

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

The main objective of the current thesis is to develop an approach that uses a topic modeling technique to cluster Arabic documents. A recent algorithm, LDA2Vec has been used as a topic modeling technique.

The developed approach comprises various steps that include text collecting, text pre-processing, text representation, training and testing stage, and evaluation.

In the conducted experiments, the optimal number of topics k has been identified to be nine. Some experiments have been carried out on the dataset with split ratios of 70-30 (1890 documents for training and 810 for testing) and 80-20 (2160 documents for training and 540 for testing), with changing the values for the LDA2Vec parameters.

The proposed approach produces better results when negative sampling power $\beta = 0.75$, the number of "negative" samples $\mathbf{k} = 5$, and the highest accuracy is achieved when using pre-trained word embeddings for text representation. Furthermore, the proposed approach utilizing the LDA2Vec technique outperformed LDA, achieving an accuracy which 0.95 to 0.89 for LDA. In addition, the approach accomplished better results compared to a similar study as shown in Figure 4.11.

Furthermore, the proposed approach was used to identify some of the thesis topics submitted by graduate students at the University of Babylon, with the thesis title and abstract serving as input text. The proposed approach accurately clustered the theses based on their field of study.

5.2 Future Works

This thesis work is conducted to explore the advantage of using topic modeling to cluster Arabic text documents. After the implementation of the proposed approach, the following future works can be listed:

- Instead of relying on a unigram to produce word embeddings vectors, it may be worthwhile to consider producing these vectors by extracting n-grams words from the training corpus.
- Automatically set the number of topics.

REFERENCES

[1] R. Albalawi, T. H. Yeap, and M. Benyoucef, "Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis," *Front. Artif. Intell.*, vol. 3, no. July, pp. 1–14, 2020, doi: 10.3389/frai.2020.00042.

[2] E. For, "C LUSTERING W EB S EARCH R ESULTS FOR," vol. 2, no. 2, pp. 17–31, 2013.

[3] Q. Al-Radaideh, "Applications of Mining Arabic Text: A Review," *Recent Trends Comput. Intell.*, 2020, doi: 10.5772/intechopen.91275.

[4] C. H S and M. K. Shenoy, "Advanced text documents information retrieval system for search services," *Cogent Eng.*, vol. 7, no. 1, pp. 0–16, 2020, doi: 10.1080/23311916.2020.1856467.

[5] I. Vayansky and S. A. P. Kumar, "A review of topic modeling methods," *Inf. Syst.*, vol. 94, p. 101582, 2020, doi: 10.1016/j.is.2020.101582.

[6] A. Uteuov, "ScienceDirect ScienceDirect Topic model for online communities ' interests prediction Topic model for online communities ' interests prediction 8th International Young Scientist Conference on Computational Science," *Procedia Comput. Sci.*, vol. 156, pp. 204–213, 2019, doi: 10.1016/j.procs.2019.08.196.

[7] M. Alhawarat and M. Hegazi, "Revisiting K-Means and Topic Modeling, a Comparison Study to Cluster Arabic Documents," *IEEE Access*, vol. 6, no. August, pp. 42740–42749, 2018, doi: 10.1109/ACCESS.2018.2852648.

[8] F. Yi, B. Jiang, and J. Wu, "Topic Modeling for Short Texts via Word Embedding and Document Correlation," *IEEE Access*, vol. 8, pp. 30692–30705, 2020, doi: 10.1109/ACCESS.2020.2973207.

[9] N. Computing, C. Chuan, K. Agres, and D. Herremans, “From Context to Concept: Exploring Semantic Relationships in Music with Word2Vec,” 2018.

[10] M. Alhanjouri, “Pre Processing Techniques for Arabic Documents Clustering,” *Int. J. Eng. Manag. Res.*, vol. 7, no. 2, pp. 70–79, 2017, [Online]. Available: http://www.ijemr.net/DOC/PreProcessingTechniquesForArabicDocumentsClustering.PDF.

[11] H. Almarwi, M. Ghurab, and I. Al Baltah, “A hybrid semantic query expansion approach for Arabic information retrieval,” *J. Big Data*, 2020, doi: 10.1186/s40537-020-00310-z.

[12] A. Kelaiaia and H. Merouani, “Clustering with probabilistic topic models on arabic texts: A comparative study of LDA and K-means,” *Int. Arab J. Inf. Technol.*, vol. 13, no. 2, pp. 332–338, 2016.

[13] Z. Wang, L. Ma, and Y. Zhang, “2016 IEEE First International Conference on Data Science in Cyberspace A Hybrid Document Feature Extraction Method Using Latent Dirichlet Allocation and Word2Vec,” pp. 98–103, 2016, doi: 10.1109/DSC.2016.110.

[14] F. Esposito, A. Corazza, and F. Cutugno, “Topic Modelling with Word Embeddings” *Proc. Third Ital. Conf. Comput. Linguist. CLiC-it 2016*, no. January 2018, 2016, doi: 10.4000/books.aaccademia.1666.

[15] C. E. Moody, “Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec,” 2016, [Online]. Available: http://arxiv.org/abs/1605.02019.

[16] M. Xue, “A Text Retrieval Algorithm Based on the Hybrid LDA and

Word2Vec Model," *Proc. - 2019 Int. Conf. Intell. Transp. Big Data Smart City, ICITBS 2019*, pp. 373–376, 2019, doi: 10.1109/ICITBS.2019.00098.

[17] Y. Luo and H. Shi, "Using lda2vec topic modeling to identify latent topics in aviation safety reports," *Proc. - 18th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2019*, pp. 518–523, 2019, doi: 10.1109/ICIS46139.2019.8940271.

[18] M. Hasan, M. M. Hossain, A. Ahmed, and M. S. Rahman, "Topic Modelling: A Comparison of the Performance of Latent Dirichlet Allocation and LDA2vec Model on Bangla Newspaper," *2019 Int. Conf. Bangla Speech Lang. Process. ICBSLP 2019*, no. September, pp. 27–28, 2019, doi: 10.1109/ICBSLP47725.2019.202047.

[19] P. Mishra, R. Rajnish, and P. Kumar, "A comparative study for sentiment analysis: Lda and lda2vec," *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 8, pp. 4061–4066, 2020, doi: 10.30534/ijeter/2020/06882020.

[20] K. Culmer and J. Uhlmann, "Examining LDA2Vec and Tweet Pooling for Topic Modeling on Twitter Data," *Wseas Trans. Inf. Sci. Appl.*, vol. 18, pp. 102–115, 2021, doi: 10.37394/23209.2021.18.13.

[21] C. E. Moody, "Introducing our Hybrid lda2vec Algorithm," 2016. https://multithreaded.stitchfix.com/blog/2016/05/27/lda2vec/#topic=38&lambda=1&term=.

[22] M. Daif, "Image-based Character Embedding for Arabic Document Classification," no. C, pp. 271–274, 2020.

[23] I. I. Jaber, "O VERVIEW OF THE A RABIC S ENTIMENT A NALYSIS 2021," 2021.

[24] R. K. Salem and A. E. Khder, “A survey of Arabic text classification approaches,” vol. 59, no. 3, pp. 236–251, 2019.

[25] Samar Awada, “Lebanese Arabic Institute,” 2018. https://www.lebanesearabicinstitute.com/arabic-alphabet/#:~:text=The Arabic Alphabet-,Overview,not part of the alphabet. (accessed Jan. 03, 2021).

[26] M. S. Alkoffash, “Comparing between Arabic Text Clustering using K Means and K Mediods,” no. August 2012, 2017, doi: 10.5120/8012-0675.

[27] N. Goel, “A Study of Text Mining Techniques : Applications and Issues,” no. September, 2020.

[28] M. Allahyari *et al.*, “A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques,” 2017, [Online]. Available: http://arxiv.org/abs/1707.02919.

[29] M. I. U. Haq, Q. Li, and S. Hassan, “Text Mining Techniques to Capture Facts for Cloud Computing Adoption and Big Data Processing,” *IEEE Access*, vol. 7, no. July 2021, pp. 162254–162267, 2019, doi: 10.1109/ACCESS.2019.2950045.

[30] D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural Language Processing : State of The Art , Current Trends and Challenges Natural Language Processing : State of The Art , Current Trends and Challenges Department of Computer Science and Engineering Manav Rachna International University , Faridabad-,” no. April, 2018.

[31] “An Analysis of the Applications of Natural Language Processing in Various Sectors,” pp. 598–602, 2021, doi: 10.3233/APC210109.

[32] R. Guan, H. Zhang, Y. Liang, F. Giunchiglia, and L. Huang, “Deep Feature-Based Text Clustering and Its Explanation,” vol. 14, no. 8, 2020, doi: 10.1109/TKDE.2020.3028943.

[33] A. K. Sangaiah and I. El-henawy, “Arabic text clustering using improved clustering algorithms with dimensionality reduction,” *Cluster Comput.*, vol. 4, 2018, doi: 10.1007/s10586-018-2084-4.

[34] G. Sahoo, “A hybrid approach using genetic algorithm and the differential evolution heuristic for enhanced initialization of the k - means algorithm with applications in text clustering,” *Soft Comput.*, 2018, doi: 10.1007/s00500-018-3289-4.

[35] M. Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J. & Nithya, “Preprocessing Techniques for Text Mining Preprocessing Techniques for Text Mining,” *Int. J. Comput. Sci. Commun. Networks*, vol. 5, no. October 2014, pp. 7–16, 2015.

[36] E. K. Al-Yasiri and A. Al-Azawei, “Improving Arabic sentiment analysis on social media: A comparative study on applying different pre-processing techniques,” *Compusoft*, vol. 8, no. 6, pp. 3150–3157, 2019.

[37] A. El Kah and I. Zeroual, “The effects of Pre-Processing Techniques on Arabic Text Classification,” no. February, 2021, doi: 10.30534/ijatcse/2021/061012021.

[38] A. Ayedh, G. TAN, K. Alwesabi, and H. Rajeh, “The Effect of Preprocessing on Arabic Document Categorization,” *Algorithms*, vol. 9, no. 2, 2016, doi: 10.3390/a9020027.

[39] A. M. Alayba, V. Palade, M. England, and R. Iqbal, “Arabic Language Sentiment Analysis on Health Services,” pp. 114–118,

2020.

[40] R. Mohammed, “New Arabic stemming based on Arabic patterns,” *Iraqi J. Sci.*, vol. 57, no. 3C, pp. 2324–2330, 2016.

[41] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. Abd Elaziz, and A. Dahou, “A Study of the Effects of Stemming Strategies on Arabic Document Classification,” *IEEE Access*, vol. 7, pp. 32664–32671, 2019, doi: 10.1109/ACCESS.2019.2903331.

[42] K. Darwish and D. W. Oard, “CLIR Experiments at Maryland for TREC 2002: Evidence Combination for Arabic-English Retrieval,” *Proc. 11th Text Retr. Conf.*, no. 1, pp. 703–710, 2002.

[43] Y. Li *et al.*, “Incorporating knowledge into neural network for text representation,” *Expert Syst. Appl.*, vol. 96, pp. 103–114, 2018, doi: 10.1016/j.eswa.2017.11.037.

[44] Z. Liu, Y. Lin, and M. Sun, *Representation Learning for Natural Language Processing*. 2020.

[45] E. Kazem and M. Hashim, “Arabic Sentiment Analysis for Determining Terrorism Supporters on Twitter Using Data Mining Techniques,” 2019.

[46] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi, “Sentiment analysis based on improved pre-trained word embeddings,” *Expert Syst. Appl.*, vol. 117, pp. 139–147, 2019, doi: 10.1016/j.eswa.2018.08.044.

[47] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations ofwords and phrases and their compositionality,” *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2013.

[48] U. Naseem, I. Razzak, S. K. Khan, and M. Prasad, “A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models,” *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 5, pp. 1–46, 2021, doi: 10.1145/3434237.

[49] A. El Mahdaouy, S. O. El Alaoui, and E. Gaussier, “Improving Arabic information retrieval using word embedding similarities,” *Int. J. Speech Technol.*, vol. 21, no. 1, pp. 121–136, 2018, doi: 10.1007/s10772-018-9492-y.

[50] T. Mikolov, G. Corrado, K. Chen, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” pp. 1–12.

[51] C. L. Jun, “Matrix Factorization using Window Sampling and Negative Sampling for Improved Word Representations ∗,” 2015.

[52] A. Pai, “An Essential Guide to Pretrained Word Embeddings for NLP Practitioners,” 2020. https://www.analyticsvidhya.com/blog/2020/03/pretrained-word-embeddings-nlp/.

[53] M. Mustafa, F. Zeng, H. Ghulam, and H. M. Arslan, “Urdu documents clustering with unsupervised and semi-supervised probabilistic topic modeling,” *Inf.*, vol. 11, no. 11, pp. 1–16, 2020, doi: 10.3390/info11110518.

[54] N. Abed, A. Shanan, H. A. Lafta, and S. Z. Al Rashid, “Bacteria taxonomic classification using Machine- learning models,” no. January, 2021.

[55] W. Zhao *et al.*, “A novel procedure on next generation sequencing data analysis using text mining algorithm,” *BMC Bioinformatics*, pp.

1–16, 2016, doi: 10.1186/s12859-016-1075-9.

[56] W. Wang, Y. Feng, and W. Dai, “Topic analysis of online reviews for two competitive products using latent Dirichlet allocation,” *Electron. Commer. Res. Appl.*, vol. 29, pp. 142–156, 2018, doi: 10.1016/j.elerap.2018.04.003.

[57] T. Gupta and S. P. Panda, “Clustering Validation of CLARA and K-Means Using Silhouette DUNN Measures on Iris Dataset,” *Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019*, pp. 10–13, 2019, doi: 10.1109/COMITCon.2019.8862199.

[58] M. N. R, J and P. R, “Performance Analysis of Text Classification Algorithms using Confusion Matrix,” *Int. J. Eng. Tech. Res.*, vol. 0869, no. 4, pp. 75–78, 2016, [Online]. Available: http://www.erpublication.org/published_paper/IJETR042741.pdf.

[59] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, “Hybrid clustering analysis using improved krill herd algorithm,” *Appl. Intell.*, vol. 48, no. 11, pp. 4047–4071, 2018, doi: 10.1007/s10489-018-1190-6.

[60] L. M. Abualigah, A. T. Khader, and M. A. Al-Betar, “Multi-objectives-based text clustering technique using K-mean algorithm,” *Proc. - CSIT 2016 2016 7th Int. Conf. Comput. Sci. Inf. Technol.*, pp. 4–9, 2016, doi: 10.1109/CSIT.2016.7549464.

[61] I. Mokriš and L. Skovajsová, “Comparison of two document clustering techniques which use neural networks,” *ICCC 2008 - IEEE 6th Int. Conf. Comput. Cybern. Proc.*, pp. 75–78, 2008, doi: 10.1109/ICCCYB.2008.4721382.

[62] J. Qiang, Y. Li, Y. Yuan, and X. Wu, “Short text clustering based on Pitman-Yor process mixture model,” *Appl. Intell.*, vol. 48, no. 7, pp. 1802–1812, 2018, doi: 10.1007/s10489-017-1055-4.

[63] M. Jahangoshai Rezaee, M. Eshkevari, M. Saberi, and O. Hussain, “GBK-means clustering algorithm: An improvement to the K-means algorithm based on the bargaining game,” *Knowledge-Based Syst.*, vol. 213, p. 106672, 2021, doi: 10.1016/j.knosys.2020.106672.

[64] D. Abuaiadah, J. El Sana, and W. Abusalah, “On the Impact of Dataset Characteristics on Arabic Document Classification,” *Int. J. Comput. Appl.*, vol. 101, no. 7, pp. 31–38, 2014, doi: 10.5120/17701-8680.

[65] “Wiki word vectors,” *fasttext.cc*, 2020. https://fasttext.cc/docs/en/pretrained-vectors.html.

Appendix A

The Published Paper

Formal Acceptance

2nd International Conference of Information Technology to enhance E-learning and other Application-2021

[IT-ELA 2021]

To/ Doaa Wahhab Alkhafaji, Sura Al-Rashid

Dear respected author(s):

With heartiest congratulations. We are pleased to inform you that based on the recommendations of the reviewers and the Technical Committees, your paper entitled:

"A Topic Modelling for Clustering Arabic Documents"

It has been accepted for oral presentation at the 2nd International Conference of Information Technology to enhance E-learning and other application-2021[IT-ELA 2021], technically sponsored by IEEE, to be held on Dec 28-29, 2021, Baghdad, Iraq.

Your paper will be submitted (**within the conference proceeding**) to the IEEE Xplore digital library for (**final acceptance of uploading to the Digital library**).

IT-ELA 2021 Baghdad – Iraq

Email:it-ela2021@baghdadcollege.edu.iq

Website: https://baghdadcollege.edu.iq/it-ela2021

Baghdad College of Economic Sciences University

Computer Sciences Department

ID: 1570774625

2nd International Conference of Information Technology to enhance E-learning and other Application, (IT-ELA2021),
Baghdad college of economic sciences University, Baghdad, Iraq

A Topic Modeling for Clustering Arabic Documents

Doaa Wahhab Alkhafaji
College of Information Technology
University of Babylon
Hilla,Iraq
duaa.alkhafaje@gmail.com

Sura Al-Rashid
College of Information Technology
University of Babylon
Hilla,Iraq
sura_os@itnet.uobabylon.edu.iq

2021 2nd Information Technology To Enhance e-learning and Other Application (IT-ELA) | 978-1-6654-0952-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/IT-ELA52201.2021.9773538

***Abstract*—Topic modeling is a type of statistical data mining technique for discovering the abstract "topics" that occur in a collection of articles or documents and the most widely used topic modeling technique is LDA. In our paper, we tested the effectiveness of a recently developed topic modeling approach (LDA2Vec) that was introduced by Chris Moody with our Arabic dataset. LDA2Vec is a hybrid approach of LDA and a highly popular word-embedding model (Word2Vec). Our goal is to find a method for automatically clustering Arabic documents by topic and categorizing them for use in a recommendation system and searching. The performance of the model was evaluated using a corpus of Arabic documents divided into nine categories. Despite the grammatical variations between Arabic and English, the model worked well with the Arabic language when it was implemented, as we observed in our study. As a conclusion of our findings, LDA2Vec gave (82.40%) accuracy over topics for test documents, which is greater than LDA accuracy (67.96%), which was evaluated with the same dataset.**

Keywords*— *Arabic Topic Modeling, Text mining, LDA, LDA2Vec, Clustering.

I. Introduction

In recent years, the majority of information on the Internet is represented as texts. The amount of electronic text available on the web is rapidly increasing because of the introduction of online blogs, newspapers, and social networking sites; this presents a significant challenge in terms of information retrieval and extracting the relevant knowledge that is required. Developing efficient strategies or tools for searching, indexing, and organizing enormous amounts of data became a necessity. Topic modeling is one of the strategies for extracting documents' hidden meaning, as well as categorizing and analyzing text data automatically [1][2].

Topic modeling is a type of statistical model and a form of text mining used to arrange large collections of documents into a smaller number of abstract "topics". Topic modeling has gotten a lot of attention in the field of study in the previous several years. It is employed in a variety of applications, including information retrieval (IR) and natural language processing (NLP).

Clustering is the most important unsupervised learning task; as with all other problems of this type, it involves identifying a structure in a set of unlabeled data, similar to topic modeling [3].

Topic modeling and clustering are approaches that require a number of categories to be defined in advance but no labels [4]. LDA is a probabilistic topic model that identifies latent topics in a large number of documents and assigns a probability distribution to each document based on the discovered topics [5]. Word2vec is a Neural Network (NN) model that uses a vast corpus of text to learn word associations. To determine semantic similarity, it calculates the cosine resemblance between the word vectors. Words with similar meanings have similar vectors, while words with different meanings have different vectors [6]. LDA2Vec is a hybrid technique of LDA and Word2Vec implemented by Chris Moody [7]. LDA2Vec implements both words and topics into a single framework. LDA2Vec result is a set of sparse document weight vectors, as well as easily interpretable topic vectors. Although the performance of LDA2Vec is similar to traditional LDA, using automatic differentiation methods makes the method scalable to the vast datasets.

In this paper, we applied the LDA2Vec technique to the Arabic corpus, which is one of the most commonly used and spoken languages in the world, with over 422 million people using it. Arabic documents became increasingly common in electronic form, so the need for clustering documents became very necessary. Because of the unique morphological principles of the Arabic language, there are few studies in the literature on the retrieval or mining of Arabic electronic text documents [8].

We used a corpus of 2700 documents gathered via scanning well-known and trustworthy Arabic websites, with different categories such as Art, Literature, Religion, Politics, and others.

This paper is divided into six sections. The second section reviews and discusses relevant studies in this domain. Section three provides a quick overview of LDA, LDA2Vec, and the topic extraction method from a corpus. Section four presents the experiment design and processing techniques for training the. The findings of the experiment are provided in section five. In section six, we summarize the results of our experiment and make recommendations for further research.

II. Related Work

A few research papers deal with topic modeling utilizing the LDA2Vec technique and apply it to document texts in English and other languages. As far as we know, this is the first study that applies topic modeling using the LDA2Vec technique for clustering Arabic text documents.

C.Moody (2016) [7] proposed a modified LDA2Vec model that is a combination of the LDA and the word2vec models. He trained his model using the 20Newsgroups dataset and the Hacker News Comments corpus. As a result, the method

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات
قسم البرمجيات

عنقدة الوثائق العربية بأستخدام LDA2Vec لنمذجة المواضيع

أطروحة

مقدمة الى مجلس كلية تكنولوجيا المعلومات للدراسات العليا بجامعة بابل والتي هي جزء من متطلبات نيل درجة الماجستير في فلسفة تكنولوجيا المعلومات – قسم البرمجيات

من قبل

دعاء وهاب ابراهيم هدوان

بإشراف

أ.م.د سرى زكي ناجي علوان

1444هـ

2022م

الــخلاصــة

بسبب الزيادة الهائلة في عدد الوثائق النصية العربية المتاحة على الإنترنت وفي قواعد البيانات ، يواجه الباحثون تحديًا كبيرًا في إيجاد طرق أفضل للتعامل مع كمية كبيرة من البيانات. لذلك ، أصبح من الضروري تطوير تقنيات أو أدوات فعالة للمساعدة في اكتشاف وتحليل المعلومات في الوثائق العربية. يُعد تجميع الوثائق باللغة العربية جانبًا مهمًا من جوانب توفير التنقل التخميني وتقنيات التصفح من خلال تنظيم كميات هائلة من البيانات في عدد صغير من المجموعات المحددة.

في هذه الأطروحة ، تم تصميم نهج يستخدم نمذجة الموضوع كتقنية لعنقدة الوثائق العربية. تم استخدام خوارزمية النمذجة الموضوعية التي تم تطويرها مؤخرًا ، LDA2Vec في هذا النهج. LDA2Vec هي خوارزمية هجينة قدمها كريستوفر مودي في عام 2016 ، والتي تنفذ كل من الكلمات والموضوعات في إطار عمل واحد. يجعل LDA2Vec كميات كبيرة من النصوص ذات قيمة للأشخاص (بدلاً من الأجهزة) مع تسهيل تعديل النموذج. نتائج LDA2Vec عبارة عن مجموعة من متجهات وزن المستندات المتناثرة ، بالإضافة إلى متجهات الموضوع التي يسهل تفسيرها.

يتكون النموذج من عدة مراحل وهي جمع الوثائق النصية ، المعالجة المسبقة للنصوص، وتمثيل النصوص ، ومرحلة التدريب باستخدام خوارزمية LDA2Vec ، ومرحلة الاختبار ، وتقييم النموذج. تم اختبار النموذج المطور باستخدام مجموعة بيانات إخبارية عربية مستخدمة في دراسات سابقة مماثلة. أظهرت النتائج أن نموذج LDA2Vec متفوق من حيث جودة العنقدة للوثائق النصية العربية وفقًا لمقاييس خارجية مثل النقاء ومقياس F والدقة وغيرها من المقاييس. يتضح في هذه الأطروحة أن نقاء النموذج المطور هو 0.88 مقارنة بـ 0.75 لـ Latent Dirichlet Allocation (LDA)، وهي أحد أكثر تقنيات النمذجة الموضوعية استخدامًا ، وهذه النتائج أعلى مقارنة بدراسة حديثة مماثلة.