



Automatic text classification using bag of words and bag of concepts based representations

Alaa Alahmadi

Publication date

01-01-2016

Licence

This work is made available under the [CC BY-NC-SA 1.0](#) licence and should only be used in accordance with that licence. For more information on the specific terms, consult the repository record for this item.

Document Version

1

Citation for this work (HarvardUL)

Alahmadi, A. (2016) 'Automatic text classification using bag of words and bag of concepts based representations', available: <https://hdl.handle.net/10344/5224> [accessed 30 Aug 2022].

This work was downloaded from the University of Limerick research repository.

For more information on this work, the University of Limerick research repository or to report an issue, you can contact the repository administrators at ir@ul.ie. If you feel that this work breaches copyright, please provide details and we will remove access to the work immediately while we investigate your claim.



UNIVERSITY of LIMERICK

OLLSCOIL LUIMNIGH

Department of Electronic and Computer Engineering

Automatic Text Classification
using Bag of Words and Bag of Concepts Based
Representations

Submitted by

Alaa Alahmadi

For the Award of Doctor of Philosophy

Supervised by

Dr. Abdulhussain E.Mahdi

Submitted to the University of Limerick

18 May 2016

Abstract

Automatic Text Classification using Bag of Words and Bag of Concepts Based Representations

Alaa Alahmadi

Automatic Text Classification (ATC) is one of the most important tasks in data mining for organizing information and knowledge discovery. The goal of ATC is to alleviate the need of manually organizing large collections of text documents, which is done by assigning one or more predefined categories to a given textual document via applying appropriate natural language processing techniques. Overall, the classification process involves three components: text pre-processing, text representation and the classifier which is built using one of the Machine Learning (ML) algorithms. In general, all existing text representations are based on the Bag-of-Words (BOW) and Bag-of-Concepts (BOC) models and their variations. The BOW representation model ignores the semantic connections between words by breaking terms into their constituent words, and synonymous words are considered as independent words with no semantic association. The BOW limitations are addressed by using concepts as features in BOC model to represent text in ATC systems. The aim of this work is to investigate and assess the effect of communally available text representation models on the performance of ATC system, in term of the accuracy of the classification and the efficiency of implementation. To achieve that, both BOW and BOC representation models are used with the ATC system and Wikipedia as a knowledge base is utilized to provide concepts. In addition, different strategies that use both words and concepts to build combined models are reviewed and compared to BOW and BOC representation models. Moreover, two languages are used to evaluate these representation models in their ATC system, which are English and Arabic. For Arabic ATC system, different variations of BOW representation models are compared which is a result of different methods that used in text pre-processing component. Furthermore, WordNet as KBs is used to provide concepts to represent Arabic text in the ATC system. This is then followed by attempts to enrich text representation by combining the features of both BOW and BOC models, in order to further enhance the performance of the ATC. Our investigation has resulted in the development of two new strategies, namely Adding Unmapped Concepts (AUC) and Using Concepts for Terms which do not appear in the Document (CTD). Both developed strategies improve ATC systems' performance in comparison with BOW and BOC representation models. They also bring text classification to a qualitatively new level of performance when compared to other strategies. In addition, CTD developed strategy reduced the time and memory required compared to other strategies used to enrich text representation in ATC systems. The results of our experiments show that text representation is a key element affecting the performance of both English and Arabic ATC systems, and the developed strategies show improvement in both languages in ATC systems. Furthermore, using Wikipedia concepts to build BOC model for Arabic ATC shows more efficiency for representing text than BOW model which does not line with what has been stated in English ATC. The reason behind that is the complex nature of the Arabic language which contains rich morphology and a large degree of the inflections and derivations. In addition, Arabic suffers from poor a morphological tool which makes Wikipedia concepts better features to represent text.

Declaration

This thesis is presented in fulfilment of the requirements for the degree of Doctor of Philosophy. It is entirely my own work and has not been submitted to any other university or higher education institution, or for any other academic award in this university. When use has been made of the work of other people, it has been fully acknowledged and fully referenced.

Signature of Author 18 May, 2016

Alaa Alahmadi

Signature of Supervisor 18 May, 2016

Hussain Mahdi

Acknowledgment

I would like to thank my supervisor Dr. Abdulhussain E.Mahdi, who was very helpful and encouraging. He accepted me as a Ph.D. student with little research experience and has taught me well. He introduced me to the field of data mining for which I am extremely grateful. I truly appreciate his help, guidance and patience and, more importantly, his understanding and support when I struggled with some difficulty. I would never have been able to complete this thesis without him.

I would also like to thank my friends who have been there for me to stay motivated and perform well even under pressure. I would also like to thank namely Arash Joorabchi and James Murphy for always being available whenever I knocked on their door for a small talk or having a coffee. I sincerely appreciate all Arash comments and advice on my writing and research.

I am also grateful to my Dad, Yahya Alahmadi, who was with me during my studies. Without his support and encouragement, I would never have been able to complete my work. He is a father and friend to me and to my beautiful son, Abdullah. I wish to give huge thanks to him and my Mum, who have taught me the value of a good education and for setting such good examples for me. Thank you to my parents and my brothers and sister for their love and support.

Table of Contents

Abstract	I
Declaration	II
Acknowledgment	III
Table of Contents	IV
List of Tables	VIII
List of Figures	X
List of Abbreviations	XIII
CHAPTER 1 Introduction	1
1.1 Motivation and Rationale	1
1.2 Aim and Objectives	2
1.3 Thesis Structure	3
1.4 Research Publications	5
CHAPTER 2 Automatic Text Classification: A Review	7
2.1 Introduction	7
2.2 Components of an ATC System	9
2.2.1 Text Pre-processing	10
2.2.2 Text Representation	11
2.2.3 Machine Learning algorithms for TC task	13
2.2.3.1 Feature Selection	13
2.2.3.2 Support Vector Machine classifier	16
2.2.3.3 Naive Bayes classifier	21
2.2.3.4 Decision Tree classifiers	23

2.2.3.5	Random Forest classifiers	27
2.3	Performance Evaluation	29
2.3.1	Hold-Out.....	29
2.3.2	K-fold Cross-Validation	30
2.3.3	Performance Measures	31
2.4	Summary	33
CHAPTER 3 Text Representation		34
3.1	Introduction	34
3.2	Bag of Words (BOW) Representation	34
3.3	N-gram based Representation	35
3.4	Phrase based Representation	37
3.5	Bag of Concepts (BOC) Representation	40
3.5.1	Knowledge Bases	40
3.5.2	WordNet as a Knowledge Base.....	43
3.5.3	Wikipedia as a KB.....	45
3.5.4	A Review of the BOC with Wikipedia/ WordNet.....	47
3.6	Combined words and concepts strategies.....	50
3.7	Summary	54
CHAPTER 4 Enhancing the Combined BOW & BOC Representation		56
4.1	Introduction	56
4.2	Combined Strategies	56
4.2.1	Adding Unmapped Concepts (AUC)	60
4.2.2	Using Concepts for Terms which do not appear in the Document (CTD)	69
4.3	Summary	76

CHAPTER 5 English Text Classification using Combined Representation ... 78

5.1	Introduction	78
5.2	Experimental Datasets	78
5.2.1	Classic 3 Dataset	78
5.2.2	20 Newsgroups dataset	79
5.2.3	World Wide Knowledge Base (WebKB) Dataset	80
5.3	English ATC System Design	81
5.4	Experiment Result and Evaluation	83
5.5	Discussion of Results	98
5.6	Summary	101

CHAPTER 6 Arabic Text Classification using Combined Representation .. 102

6.1	Introduction	102
6.2	Main Features of the Arabic Language	102
6.3	Experimental Datasets	107
6.3.1	Al-Khaleej Dataset	107
6.3.2	Saudi News Paper (SNP) Dataset	108
6.3.3	Arabic 1455 Dataset	109
6.4	Arabic Text Classification	109
6.5	Arabic ATC System Design	119
6.6	Experimental Result & Evaluation	122
6.6.1	Stemming Methods	122
6.6.2	Knowledge Bases	129
6.6.3	Evaluation of Combined Representation Models	131
6.7	Discussion of Results	136
6.8	Summary	141

CHAPTER 7 Conclusions	143
7.1 Summary of thesis	143
7.2 Main Findings & Contributions	144
7.3 Future work	146
References	145
Appendix A Documents Sample.....	164
Appendix B Papers Publications	168

List of Tables

Table 2.1 The contingency table of feature f and category c_j	15
Table 2.2 Training set for predicting borrowers who will default on loan payments	24
Table 2.3 Test example for an applicant with personal information to predict the category of the borrower	26
Table 2.4 Classifier possible predictions in an ATC system	31
Table 3.1 BOW text representation for the text in Figure 3.1	51
Table 3.2 BOC text representation for the text in Figure 3.1	52
Table 3.3 AC text representation for the text in Figure 3.1	52
Table 3.4 RTC text representation for the text in Figure 3.1	53
Table 3.5 ACC text representation for the text in Figure 3.1.	54
Table 4.1 Concept-Words Map for the given example.....	63
Table 4.2 Meaning of legend in scatter plot	65
Table 4.3 The classification output for the nine documents using both BOW and CTD models	74
Table 4.4 The SVM prediction for each category with the BOW and CTD models	75
Table 4.5 Classification prediction of the nine sample documents with different representation models.....	76
Table 5.1 List of categories and number of documents in the Classic 3 dataset	79
Table 5.2 List of categories and number of documents in the 20 Newsgroups dataset	80
Table 5.3 List of classes and number of documents in the WebKB dataset.....	81
Table 5.4 The performance of the SVM classifier with combined models and the BOW model to represent text in the WebKB dataset.....	90
Table 5.5 The performance of the NB classifier with combined models and the BOW to represent text in the WebKB dataset.....	91
Table 5.6 The performance of the DT classifier with combined models and the BOW to represent text in the WebKB dataset.....	92
Table 5.7 The performance of the RF classifier with combined models and the BOW to represent text in the WebKB dataset	93
Table 5.8 Average Macro F1-measure for 20 Newsgroups in different combined models using different strategies	96

Table 6.1 Example of an Arabic letter “b” (ب) and its various shapes depending on its position in the word	103
Table 6.2 List of string letters that represent prefix, suffixes, and infixes in Arabic	104
Table 6.3 Example of Arabic diacritics	105
Table 6.4 Different forms that the word (كتب) “write” can take in Arabic	106
Table 6.5 Number of documents per category in Al-Khaleej dataset.....	108
Table 6.6 Number of documents per category in SNP dataset	108
Table 6.7 Number of documents per category in Arabic 1445 dataset.....	109
Table 6.8 Summarization of different studies comparing different stemming methods in ATC for Arabic text.....	112
Table 6.9 A summary of works that used different ML algorithms in Arabic ATC system.....	114
Table 6.10 A summary of different works that used same datasets with different ML algorithms in Arabic ATC system.....	117
Table 6.11 Different representation models used in Arabic ATC	119
Table 6.12 Different datasets and their encoding	120
Table 6.13 root for the same words with different stemming method.....	139

List of Figures

Figure 2.1 The three main components of an ATC system	9
Figure 2.2 Pre-processing steps	10
Figure 2.3 The separation hyperplane.....	17
Figure 2.4 SVM and how margins are maximized	17
Figure 2.5 Separation hyperplane margin.....	19
Figure 2.6 DT for constructed based on training examples	25
Figure 2.7 Illustration of how the DT classifies a test example.....	26
Figure 3.1 Sample text	51
Figure 4.1 The three main components of an ATC system	57
Figure 4.2 Text representation component in our ATC system.....	57
Figure 4.3 Example of a document from the Classic 3 dataset.....	58
Figure 4.4 Different representation models for the document “cran.001051”	59
Figure 4.5 Process of building AUC model.....	62
Figure 4.6 The AUC representation model for the given example.....	63
Figure 4.7 The combined representation using the AUC strategy for the document “cran.001051”	64
Figure 4.8 The 602 misclassified documents of the WebKB dataset with BOW representation model in ATC system.....	66
Figure 4.9 The 953misclassified documents of the WebKB dataset with BOC representation model in ATC system.....	66
Figure 4.10 The 619 misclassified documents of the WebKB dataset with AC representation model in ATC system.....	67
Figure 4.11 The 850 misclassified documents of the WebKB dataset with RTC representation model in ATC system.....	67
Figure 4.12 The 594 misclassified documents of the WebKB dataset with ACC representation model in ATC system.....	68
Figure 4.13 The 579 misclassified documents of the WebKB dataset with AUC representation model in ATC system.....	68
Figure 4.14 process of representing document using CTD strategy.....	70
Figure 4.15 The BOW & BOC model for the documents that are belonging the dataset	71
Figure 4.16 CTD representation model for the given example	71

Figure 4.17 The combined representation using CTD strategy for the document “cran.001051”	72
Figure 4.18 Example of three documents from the Classic 3 dataset.....	73
Figure 5.1 Accuracy of the BOW and BOC with different ML algorithms in an English ATC	84
Figure 5.2 Accuracy of combined models with different ML algorithms in the Classic 3 dataset.....	85
Figure 5.3 Average accuracy with different strategies to build combined models in the Classic 3 dataset.....	86
Figure 5.4 A comparison between the BOW and combined models using the ACC and CTD strategies with Classic 3 in the ATC system.....	86
Figure 5.5 Number of features in different representation models in the Classic 3 dataset	87
Figure 5.6 Macro F1-measure of different representation models with different size of training subset of Classic 3 dataset with SVM classifier.....	88
Figure 5.7 Micro F1-measure of different representation models with different size of training subset of Classic 3 dataset with SVM classifier.....	88
Figure 5.8 The Micro F1-measure of different representation models with a different size of training subset of the WebKB dataset to train the SVM classifier	94
Figure 5.9 Performance of the SVM classifier with different representation models for the 20 Newsgroups	95
Figure 5.10 Number of features in different representation models in 20 Newsgroups.....	97
Figure 5.11 Performance of NB and RF classifier with different representation models for 20 Newsgroups	98
Figure 5.12 Average accuracy of different classifiers with BOW and BOC models in all datasets.....	99
Figure 5.13 Average Macro F1-measure of different representation models on all datasets.....	100
Figure 5.14 Average number of features in different representation models for all datasets.....	100
Figure 6.1 Algorithm for LS method	120

Figure 6.2 Classification accuracy of different BOW models with different classifiers in Arabic ATC.....	124
Figure 6.3 Number of features in representation models in different dataset.....	125
Figure 6.4 Time required by different representation models with SVM, NB, and RF classifiers.....	126
Figure 6.5 Time required by DT classifier with different BOW representation models.....	127
Figure 6.6 Average accuracy of different BOW representation models with different classifiers.....	128
Figure 6.7 Pre-processing time with different BOW representation models.....	128
Figure 6.8 Classification Accuracy of BOC-based Wikipedia and BOC-based WordNet models with different classifiers in Arabic ATC	129
Figure 6.9 Number of features of BOC-based Wikipedia and BOC-based WordNet models in different datasets	130
Figure 6.10 Time required by BOC-based Wikipedia and BOC-based WordNet models with different classifiers	131
Figure 6.11 Classification accuracy of combined models with different classifiers in the Arabic 1445 dataset.....	132
Figure 6.12 Classification accuracy of combined models with different classifiers in the SNP dataset	133
Figure 6.13 Classification accuracy of combined models with different classifiers in Al-Khaleej dataset.....	134
Figure 6.14 Number of features in the combined models with different datasets ..	135
Figure 6.15 The classification time of different combined model in the three datasets	135
Figure 6.16 Average accuracy of various BOW models in all datasets with different classifiers.....	137
Figure 6.17 Average accuracy of all datasets with different classifiers	138
Figure 6.18 Average accuracy of BOW-LS and BOC-based Wikipedia in all datasets with different classifiers	138
Figure 6.19 Average accuracy performance for combined representation models	141

List of Abbreviations

AC	Adding Concepts
Acc	Accuracy
Acc2	Accuracy2
ACC	Adding Concepts and Categories
ATC	Automatic Text Classification
ANN	Artificial Neural Networks
AUC	Adding Unmapped Concepts
AWN	Arabic WordNet
BNS	Bi-Normal Separation
BOC	Bag of Concepts
BOW	Bag of Words
BOW-LS	BOW based Light Stemming
BOW-RE	BOW based Root Extraction
CA	Classical Arabic
CBA	Classification Based Association
CTD	using Concepts for Terms which do not appear in the Document
DA	Dialectal Arabic
DB	Distance Based
DF	Document Frequency
DT	Decision Trees
FN	False Negative
FP	False Positive
FS	Feature Selection
Gss	GSS score
GSSS	GSS Square

IG	Information Gain
IR	Information Retrieval
KB	Knowledge Base
<i>k</i>-NN	<i>k</i> -Nearest Neighbor
KE	Knowledge Engineering
LS	Light Stemming
MeSH	Medical Subject Headings
MI	Mutual Information
ML	Machine Learning
MNB	Multinomial Naive Bayes
MSA	Modern Standard Arabic
MVB	Multi-Variate Bernoulli
NB	Naive Bayes
NGL	NGL coefficient
NLP	Natural Language Processing
OCR	Optical Character Recognition
OR	Odds Ratio
ODP	Open Directory Project
POS	Parts of Speech
Pwr	Power
RE	Root Extraction
RF	Random Forest
RTC	Replacing Term with Concepts
SNP	Saudi News Paper
ssFCM	Semi-Supervised Fuzzy c-Means
SVM	Support Vector Machines
TC	Text Classification

TF	Term Frequency
TFIDF	Term Frequency Inverse Document Frequency
TP	True Positive
TN	True Negative
VSM	Vector Space Model
WCV	Word Cluster Vector
WebKB	World Wide Knowledge Base
WIDF	Weighted Inverse Document Frequency
χ^2	Chi-square

CHAPTER 1

Introduction

1.1 Motivation and Rationale

The exponential growth of text documents in digital form has increased the need for fast and effective navigation, browsing, and discovery of information. The branch of computer science dealing with this issue is called “data mining” and it covers a range of topics such as Information Retrieval (IR), information filtering, question answering, knowledge discovery, and Text Classification/categorisation (TC).

Text Classification (TC) is the task of assigning one or more predefined categories or classes to a given text based on its content. These categories come from a fixed set of labels, and each document may be assigned one or more categories. TC is used in many different ways from email filtering, routing news, cataloguing journal articles in digital libraries to the classification of web pages. In the early days of TC systems, human cataloguers manually assessed the main topics (categories) of each text document and organised them into meaningful structures. This was an intensive work, time consuming, and impractical in many cases with the vast volume of text documents. In response to this, Automatic Text Classification (ATC) systems were developed. The ATC does not require any human intervention or any prior knowledge of the texts and is especially useful since Machine Learning (ML) techniques have been around since the early 1990s. ATC systems automatically analyse the similarities between text documents and organise them to form a classification of texts that share similar topics.

To apply TC to a text document, each text such as a news article, scholarly publication, a document in a digital library collection, or even a fragment of a document is processed and cleaned to extract a set of weighted features with each feature corresponding to a dimension in a Vector Space Model (VSM). The traditional type of feature used to represent text in the majority of existing ATC systems are words mentioned in the text, where both positions and order of occurrence are not taken into consideration.

The model is known as a Bag of Words (BOW), and it is widely used in ATC systems because it is simple and effective. The BOW model observes and extracts information from texts in a different way to a human cataloguer which leads to a number of shortcomings.

First, the BOW model treats words independently from each other as isolated language units and ignores the connection between words that form meaningful lexical structures. Second, the BOW model ignores the fact that different words can have the same meaning (synonymy) while the same word might have different meanings in different contexts (polysemy). Third, the BOW model is limited where only words that are clearly mentioned in the training subset documents are used in the classification task. In addition, the model ignores words in the testing subset document that never appeared in the training document. Various studies attempt to overcome these shortcomings by using different types of features to build text representation models such as N-grams [1, 2], statistical phrases [3, 4], and syntactic phrases [5, 6] which use linguistically motivated features based on the syntactic information. However, these attempts had limited success because of their lack of world knowledge, and they get confused by features that cannot be detected in the training set. Other researchers used a different type of features, such as the Bag of Concepts (BOC) which use general purpose Knowledge Bases (KBs) such as Wikipedia in the text representation model. In this thesis, we explore commonly available text representation models and develop new models to enrich the text representation in our ATC system in order to enhance classification performance further.

1.2 Aim and Objectives

The aim of this work is to explore different methods to overcome the shortcomings of the BOW model and enrich text representation in the ATC systems to enhance the classification performance. This is done by pursuing two lines of research:

1. Investigating and assessing the effect of commonly available text representation models on the performance of ATC systems for both the English and Arabic language.
2. Developing new models based on combining various selected aspects of the BOW and BOC representations in order to further enhance the classification

performance in terms of the accuracy of the classification and the efficiency of implementation.

In order to achieve this aim, the following objectives have been devised:

- Develop an experimental ATC system and evaluate its performance using the basic BOW model for both English and Arabic textual datasets. For the Arabic ATC system, different variations of the BOW representation model (resulting from the application of different stemming approaches at the text pre-processing stage) are compared.
- Wikipedia as a knowledge base is used to provide concepts to represent text with the basic BOC model, and this is compared to the BOW model when used with both English and Arabic textual datasets. In addition, WordNet is used instead of Wikipedia as a knowledge base to provide concepts for an Arabic text to compare the classification performance of WordNet and Wikipedia concepts when used to build the BOC model in an Arabic ATC system. To the best of our knowledge, this has not been studied before.
- This is then followed by similar evaluations when text representation approaches based on combining both words and concepts are used, as compared to the cases of the basic BOW and BOC representations for both English and Arabic textual datasets. New models based on combining various selected aspects of the BOW and BOC representations are developed which further enhance the classification performance in terms of classification accuracy and the efficiency of implementation in the ATC system.

1.3 Thesis Structure

Chapter 2 is a review of Automatic Text Classification (ATC) systems, and it illustrates the main components of building an efficient ATC system. First, the techniques used to transform text documents into a form that is suitable for automatic processing is discussed in the text pre-processing stage. In this chapter, the different techniques for feature selection and their impact on the TC task are reviewed which leads to a discussion about how to represent text using the best features from the text pre-processing component. The different weighting schemes for features are reviewed as are the most common Machine Learning (ML)

algorithms used in ATC systems. This chapter concludes with a discussion about the different measures that evaluate the performance of an ATC system and the different methods that divide a set of documents into training and testing subsets.

Chapter 3 offers a detailed explanation of the text representation component and how text representation is an important aspect of a successful ATC system. We also discuss how using different types of features in text representation can have a significant impact on the performance of ATC systems. In this chapter we show the different types of features used in literature, in text mining in general, and TC in particular. These features include single words, N-grams, statistical phrases, and syntactic phrases which are described, and both their advantages and disadvantages are discussed. Furthermore, we consider concepts which provide a unique meaning and behave as a unit of knowledge used as features for text representation models. This chapter looks at the different characteristics of Knowledge Bases (KBs) in general and WordNet and Wikipedia in particular and how concepts are identified in the free text.

Chapter 4 introduces two new strategies that use both words and concepts to enhance the classification performance in terms of the accuracy of the classification and the efficiency of implementation. The first is Adding Unmapped Concepts (AUC), and the second is using Concepts for Terms which do not appear in the Document (CTD). For both developed text representation strategies, the design of these proposed strategies is reviewed, and a document example is given to show how they enrich the text representation in the VSM. The text representation combined models that are constructed using these proposed strategies are compared to the basic BOW and BOC representation models and to other strategies previously used to build text representation combined models.

Chapter 5 describes in detail the three English datasets used to evaluate the proposed developed strategies with English text in an ATC system in comparison to standard BOW and BOC models. The experiment designs are clearly laid out where both a hold-out method and cross-fold validations in conjunction with different ML algorithms are used. In addition, a different size of training dataset is used to analyse the impact of the size of the text representation model on the performance of the ATC system. We also look at the comparison between existing and developed strategies which use both words and concepts to enhance the

classification performance in terms of the accuracy of the classification and the efficiency of implementation.

Chapter 6 considers the question of the Arabic language and its complexity in comparison to other languages such as English. The three Arabic datasets used in this thesis are introduced, and our reasons for using these datasets are discussed. The most common studies on Arabic ATC systems are reviewed from different points of view, namely the impact of the pre-processing stage, different types of features used in the text representation component, and applying different classification algorithms and comparing them with Arabic text in the ATC system. The experimental design is clearly illustrated, and the impact of the morphological analysis is investigated in regards to the classification accuracy of the Arabic ATC system, the size of the representation models, and the time required by ML algorithms to train data and build a model to classify test data. In addition, concepts as features to represent Arabic text are used in the ATC system, and two KBs are used to provide concepts. Both developed strategies and existing strategies that combine both words and concepts from Wikipedia are applied to the Arabic ATC system and compared.

Chapter 7 concludes the work of this thesis and presents a summary of each chapter discussing the main findings of the experiments and possible future work.

1.4 Research Publications

- A. Alahmadi, A. Joorabchi, and A. E. Mahdi, “Arabic Text Classification using Bag-of-Concepts Representation,” in *Proc. 6th International Conf. Knowledge and Discovery Information Retrieval*, Rome, Italy, 2014, pp.374-380.
- A. Alahmadi, A. Joorabchi, and A. E. Mahdi, “Combining Bag-of-Words and Bag-of-Concepts Representations for Arabic Text Classification,” in *Proc. 25th IET Irish Signals and Systems Conf.*, and *China-Ireland Int. Conf. Information and Communications Technologies*, Limerick, Ireland, 2014, pp.343-348.

- A. Alahmadi, A. Joorabchi, and A. E. Mahdi, “A New Text Representation Scheme Combining Bag-of-Words and Bag-of-Concepts Approaches for Automatic Text Classification,” in *Proc. 7th IEEE GCC Conf. and Exhibition*, Doha, Qatar, 2013, pp.108-113.

Copies of above papers are provided in Appendix B.

CHAPTER 2

Automatic Text Classification: A Review

2.1 Introduction

This chapter introduces Automatic Text Classification (ATC), and it discusses the main components for building an ATC system namely, text pre-processing, text representation, and the most common Machine Learning (ML) algorithms (classifiers) used for TC.

TC is one of the most important tasks in data mining for organising information and knowledge discovery. Data mining includes applications that involve the process of detecting meaningful information or potentially useful patterns hidden in a large amount of data using ML and Natural Language Processing (NLP) techniques. The goal of ATC is to alleviate the need for manually organising large collections of text documents [7]. It is used in different applications such as email filtering, routing news, digital libraries, and the classification of web pages.

Text or document classification is the task of assigning one or more predefined categories to a given textual document via applying appropriate NLP techniques. More formally, if d_i is a document from a given set of documents D , and $C = \{c_1, c_2, \dots, c_j\}$ is a set of predefined categories, then the task of the classifier is to assign one or more categories, c_j , to the document d_i [8].

Based on the number of categories that should be assigned to the document, TC can be divided into three groups:

- Binary classification: in this case the set of available categories has only two categories and the document is assigned to precisely one category.
- Single label classification: in this case the set of available categories is more than two, and the document is assigned to only one category.
- Multi label classification: in this case the set of available categories is more than two and the document could be assigned to multiple categories.

Based on the relationship between the available categories in multi label classification, it can also be divided into two types:

- Flat classification: in this case the set of available categories are isolated, and there is no structure that defines the relationships between them.
- Hierarchical classification: in this case the set of categories are related, and they have a structure that defines the relationships between categories where main categories are divided into sub-categories.

In a complete ATC system a decision needs to be made for each pair of documents d_i and a category c_j ; based on this decision, TC can be divided into two groups:

- Hard classification: in this case the decision of ATC system should be either “true” or “false” which indicates membership or non-membership of a given category.
- Soft classification: in this case the ATC system provides a numeric score as the decision. This number is in a specific range which is used to measure the degree of the classifier’s confidence that a document d_i belongs to a category c_j .

Work on ATC started in the 1960s and Maron's M.E research on probabilistic indexing is considered as the first work in the field [9]. Until the late 1980s, the most common approach for TC was Knowledge Engineering (KE) which is based on manually defining a set of rules that encode expert knowledge to specify how to classify text documents according to the given categories. A famous example is the CONSTRUE system that was developed for Reuters [10]. This approach is very time consuming as the rules are manually defined by a knowledge engineer with the assistance of a domain expert. Moreover, such rules cannot be easily reused across domains or even across datasets from the same domain. Consequently, since the early 1990s ML techniques have been introduced to the TC problem as a result of the increasing availability of more powerful hardware especially high processing computers with memory storage capable of handling thousands of documents.

Once an ML-based classifier is built (by learning from a training set of pre-classified documents) it will be able to classify a test set of unclassified documents. Using ML algorithms for TC has been shown in many cases to yield an accuracy comparable to that achieved by human experts [8]. ML algorithms offer

considerable savings in terms of human experts' time since no involvement from either knowledge engineers, or domain experts are needed to build the classifier.

To understand the different components of an ATC system and the necessary steps to assign a document to a category, a full description of an ATC system and its components is provided in the next section.

2.2 Components of an ATC System

Normally the development of an ML-based ATC system involves two phases, a training phase, and a testing phase, as illustrated in Figure 2.1. In each phase, the actual classification system involves three main components: text pre-processing, text representation, and the classifier which is built using a common Machine Learning (ML) algorithm. The classifier from the training phase is built by learning from labelled documents, and it is going to be used in the testing phase to classify unlabelled documents as shown below.

Overall, the classification process begins by passing all the documents in the dataset through a text pre-processing component where their textual content is cleaned and processed. The result of this component is a set of well-defined features. These features are then used by the text representation component.

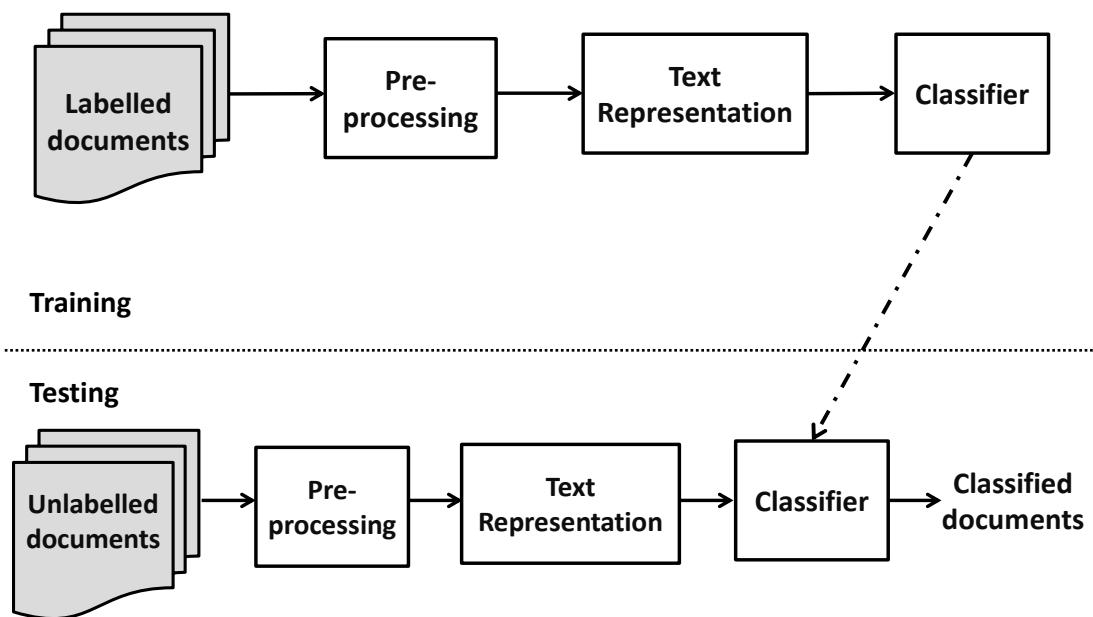


Figure 2.1 The three main components of an ATC system

Finally, the outputs of these components are fed to ML based classification algorithms (classifiers). The following sections describe each of the main components of an ATC system in more detail.

2.2.1 Text Pre-processing

Pre-processing is defined as the task of converting text to a well-defined set of features. These features could be words, concepts, or a combination of both which are used to represent text in ATC systems. In this component documents are cleaned and processed by removing noise and irregularities which negatively affect the classification performance. This includes the removal of all digits, punctuation marks, stop words, and common words such as pronouns and prepositions. Figure 2.2 illustrates the four steps of pre-processing.

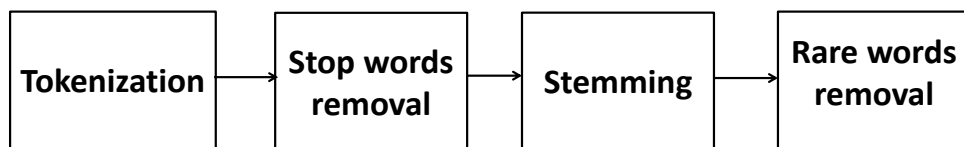


Figure 2.2 Pre-processing steps

1. **Tokenization:** In this step the text is broken up into individual and meaningful units known as tokens. Each token is separated from others by a particular character or symbol. For example, in written English, words are separated by a space and each word is considered as a single meaningful unit. The remaining punctuation marks, digits, numbers and dates are removed as they are considered as noise and irregularities which do not carry any value for the classification task.
2. **Stop words removal:** In this step, common words such as pronouns, prepositions, and articles are removed from the text. These words are called “stop words” and they occur frequently. They do not have any discriminatory significance for classification.
3. **Stemming:** In this step, all words are reduced to their root form where morphological information is used to merge various word forms such as plurals and verb conjugations into their distinct roots. For example, the words “write”, “writing”, and “writer” can be reduced to their root “write”. Overall, stemming

is an important step in NLP, text mining, and information retrieval related applications. In the case of high dimension feature space problems, stemming could be used as a dimension reduction technique to improve the performance of ATC systems. The most common algorithm for stemming English text is Porter's algorithm, and it has repeatedly been shown to be empirically effective [11].

4. Rare words removal: Generally, the remaining number of words is still large after removing stop words and stemming. Most of these words rarely appear and produce a complex text representation model. In addition, words that appear only occasionally are hard to learn from in classification because they do not discriminate between documents. Removing these words based on a threshold (e.g., two or three) improves the classification learning speed. Removing rare words is the simplest method among features selection methods (more details in Section 2.2.3.1).

Having completed the pre-processing the resulting features are used in the next component of the ATC system which is the text representation component.

2.2.2 Text Representation

The traditional approach to represent text is by using a Vector Space Model (VSM) [12]. This model transforms textual data into a weighted high dimensional vector, each dimension corresponding to a feature. These features are the result of the pre-processing phase. These features could be single words, phrases, concepts, or a combination of words and concepts (more details in Chapter 3).

Let D be a set of documents, and F the set of features used to represent these documents. Each document, $d_i \in D$, is expressed as a weighted high dimensional vector $\vec{d}_i$ and each dimension corresponds to a unique feature, such that $\vec{d}_i = (f_1, f_2, f_3, \dots, f_m)$ where $f_m \in F$ is the weight of a given feature of that document.

Different methods are used to weight features in ATC systems, namely:

- Boolean weighting: This is the simplest way for weighting the features. If a feature f_m occurs in a document d_i at least once then, the weight of f_m will take the value of one. Otherwise, it will take the value of zero [13, 14].

- Term Frequency (TF): In the TF weighting scheme, the weight of a feature f_m in the VSM is the number of times that the feature appears in a special document d_i [13, 14].
- Term Frequency Inverse Document Frequency (TFIDF): The TFIDF is a statistical weighting scheme proposed by Salton and Buckley [15]. It is a common method used in the TC problems and other related applications in data mining. It is a simple and effective method for weighting features in text documents for classification purposes [7, 13, 16]. The weight of a feature f_m in a document d_i using TFIDF is defined as:

$$TFIDF(f_m, d_i) = TF(f_m, d_i) \cdot IDF(f_m) \quad (2.1)$$

where $TF(f_m, d_i)$ is the frequency of a feature, $f_m \in F$, F is the set of all features that are considered in the representation model. $TF(f_m, d_i)$ is defined as:

$$TF(f_m, d_i) = \frac{n(f_m, d_i)}{|d_i|} \quad (2.2)$$

where $n(f_m, d_i)$ is the occurrence frequency of the feature f_m in document d_i , normalized by $|d_i|$ which is the length of d_i . The Inverse Document Frequency $IDF(f_m)$ is defined as:

$$IDF(f_m) = \log \frac{|D|}{DF(f_m)} \quad (2.3)$$

where $IDF(f_m)$ returns the inverse document frequency of the feature f_m . It counts the number of documents in which the feature f_m appears, $|D|$ is the total number of documents in the dataset. The $IDF(f_m)$ parameter has the effect of reducing the weight of those words which appear in a large portion of the dataset.

After extracting features from the text in the pre-processing phase, the features are weighted using one of the above methods. The features are then used to build a VSM model for each document in the dataset. The VSM representation model is fed into an ML classification algorithm to build and learn a classifier model from labelled documents to predict the category of unlabelled documents.

2.2.3 Machine Learning algorithms for TC task

In the last twenty years, a wide range of ML algorithms have been used for ATC. However, the most popular and successful ML algorithms which are frequently used for TC are Support Vector Machines (SVM), Naive Bayes (NB), Decision Trees (DT), and Random Forest (RF). To use these ML algorithms with more efficiency, Feature Selection (FS) techniques are used to select the best features to represent the text. The following section provides a review of the FS techniques and the above ML algorithms.

2.2.3.1 Feature Selection

For TC, Feature Selection (FS) can be defined as the task of finding features which are more significant than others in the classification problem and discard other features that are irrelevant and redundant. FS was first introduced and investigated in the early 1960s by Marill and Green [17] and have been developed for the field of pattern recognition. In general, most learning algorithms are not designed to deal with high dimensions of feature space and by using FS techniques the dimensionality of the feature space is reduced. In addition, FS techniques enable learning algorithms to operate faster and more effectively. In some cases, the accuracy of learning algorithms can be improved using feature selection. Some of the common FS techniques include:

- Document Frequency (DF) is defined as the number of documents in which a feature occurs. The idea of DF is like the rare words removal (see Section 2.2.1) where features are removed based on their document frequency. In other words, the value of a specific feature is the number of documents containing that feature in the training dataset. Then, all features where their document frequencies are less than a predefined threshold are removed from the representation model. DF is one of the simplest techniques for FS, which has been shown to yield a similar performance to that of more computationally involved FS techniques, such as information gain and Chi-square (χ^2) [18]. DF assumes that features with low document frequency do not give a good prediction of a document category and/or have an insignificant effect on the performance of the classification [18, 19].

- Information Gain (IG) is frequently employed as a term goodness criterion in the field of ML [18, 20]. It measures the number of bits of information obtained for category prediction by knowing the presence or absence of a feature in a document from the training subset [18]. Let $C = \{c_1, c_2, \dots, c_j\}$ denote the set of categories in the dataset. The IG for a given feature f is computed as:

$$\begin{aligned} \text{IG}(f) = & - \sum_{j=1}^m P(c_j) \log P(c_j) \\ & + P(f) \sum_{j=1}^m P(c_j|f) \log P(c_j|f) \\ & + P(\bar{f}) \sum_{j=1}^m P(c_j|\bar{f}) \log P(c_j|\bar{f}) \end{aligned} \quad (2.4)$$

where m is total number of categories,

- $P(c_j)$ is the probability of category c_j .
- $P(c_j|f)$ is the probability of category c_j if f is in the document, i.e., which proportion of the documents where f occurs belongs to the category.
- $P(c_j|\bar{f})$ is the probability of category c_j if f is not in the document i.e., which proportion of the documents where f does not occur belongs to the category.
- $P(f)$ is the probability of feature f appearing in a given document in the dataset.
- $P(\bar{f})$ is the probability of feature f not appearing in a given document in the dataset.

For each feature in the training dataset, the IG is computed, and the features with an IG below a predefined threshold are removed from the representation model.

- Mutual Information (MI) is a method commonly used in statistical language modelling of word association and related applications [21, 22]. It determines the mutual dependency between a feature f and a category c_j .

Using the two way contingency Table 2.1, A denotes the number of times that f appears in c_j , B denotes the number of times that f appears without c_j , E denotes the number of times that c_j appears without f , K denotes the number of times that neither c_j nor f are appeared and N is the total number of documents,

Table 2.1 The contingency table of feature f and category c_j .

	c_j	NOT c_j	Total
f	A	B	$A+B$
NOT f	E	K	$E+K$
Total	$A+E$	$B+K$	N

then the MI criterion between f and c_j is computed as:

$$MI(f, c_j) = \log \frac{P(f \wedge c_j)}{P(f)P(c_j)} \quad (2.5)$$

moreover, is estimated using:

$$MI(f, c_j) \approx \log \frac{AN}{(A+E)(A+B)} \quad (2.6)$$

If f and c_j are independent then the MI (f, c_j) has a natural value of zero. To measure the goodness of a feature in a global feature selection, the category specific score of a feature is combined in two alternate ways:

$$MI_{average}(f) = - \sum_{j=1}^m P(c_j) MI(f, c_j) \quad (2.7)$$

$$MI_{maximum}(f) = \max_{j=1} \{MI(f, c_j)\} \quad (2.8)$$

where m , is the number of categories in the training dataset. A weakness of MI is that the scores are not comparable across features with broadly different frequencies. This is because the marginal probabilities of features strongly affect the MI scores, and as a result rare features will have higher scores than common features.

- Chi-square (χ^2) is a method used to determine the degree of dependency between a feature f and a category c_j [18]. The higher the value of χ^2 , the higher the dependency or association between feature f and a category c_j using the two way contingency Table 2.1 of a feature f and a category c_j , where A denotes the number of times that f appears in c_j , B denotes the number of times that f appears without c_j , E denotes the number of times that c_j appears without f , K represents

the number of times that f and c_j do not appear, and N is the total number of documents. The χ^2 of feature f in relation to category c_j is computed as:

$$\chi^2(f, c_j) = \frac{N(AK - EB)^2}{(A + E)(B + K)(A + B)(E + K)} \quad (2.9)$$

If χ^2 is equal to zero, that means the feature f and the category c_j are independent [22]. To measure the goodness of a feature globally, for each category, the χ^2 between each unique feature in the training dataset and that category is computed, then the category specific scores for the feature are combined using the following functions [21, 22]:

$$\chi^2_{average}(f) = \sum_{j=1}^m P(c_j) X^2(f, c_j) \quad (2.10)$$

$$\chi^2_{maximum}(f) = \max_{j=1} \{X^2(f, c_j)\} \quad (2.11)$$

where m , is the number of categories in the training dataset.

Unlike MI, the χ^2 is a normalized value, and it is comparable across features under the same category. Both χ^2 and MI are not reliable for low frequency features [22, 23].

2.2.3.2 Support Vector Machine classifier

The Support Vector Machine (SVM) is supervised learning algorithm that analyses data and recognises patterns. It is based on the structural risk minimisation principle [24] from computational learning theory. SVM was first introduced by Vapnik in 1995 for solving two-category pattern recognition problems [24]. The SVM was adopted for the problem of TC by Joachims in 1998 [25] and subsequently used by others [26, 27].

The SVM is defined over a vector space where the problem is to find a decision surface, or hyperplane, that separates the data points to two categories. As shown in Figure 2.3 in order to define the greatest separation dividing the data into two groups, we need to introduce a margin between the two categories.

Let D be a training set of documents where each document belongs to one of two categories. The SVM classifier builds a model that predicts whether a new document falls into one category or the other.

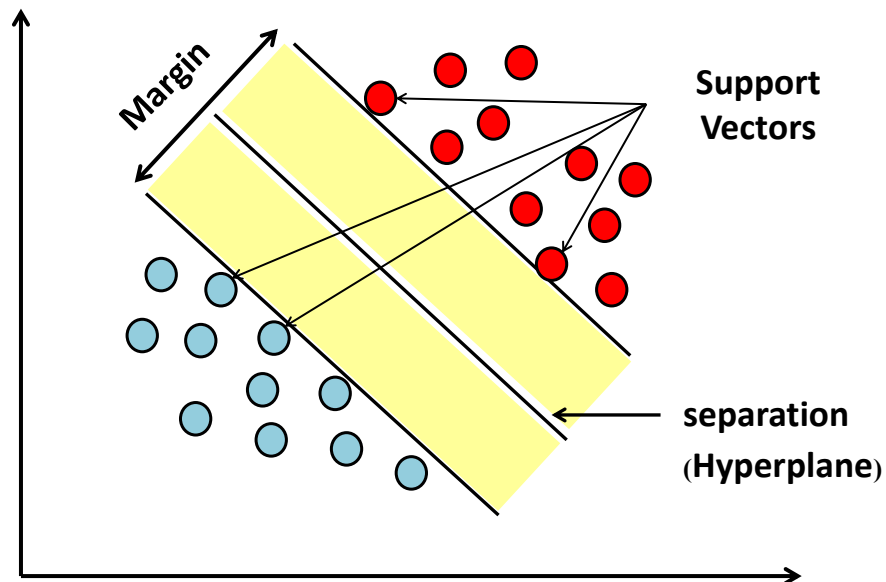


Figure 2.3 The separation hyperplane

Moreover, the SVM model is a representation of documents as points in space, so that documents belonging to one category can be separated from others by a clear gap, or margin as it is otherwise known, and it should be as wide as possible. A new document is predicted to belong to a category based on which side of the gap it falls on [7, 28, 29]. The SVM problem is to find the decision surface that maximizes the margin between the data points in a training dataset (see Figure 2.4).

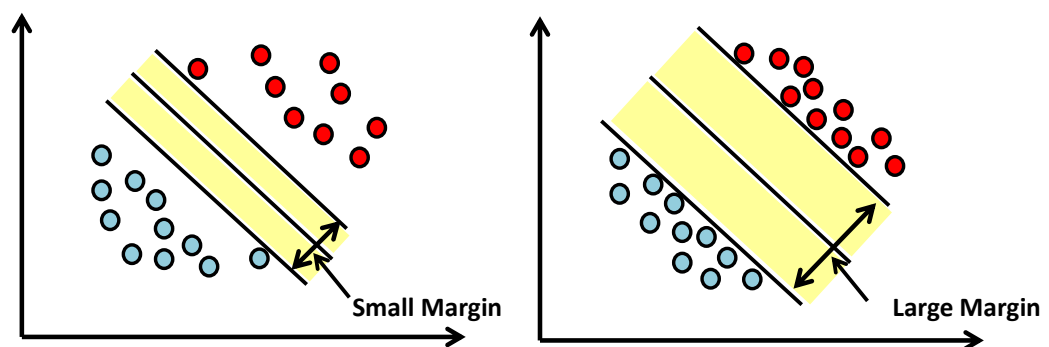


Figure 2.4 SVM and how margins are maximized

As can be seen in Figure 2.4, a good separation is achieved by the largest margin without causing misclassification of the data [7, 28, 30].

The separation hyperplane in dimensional space can be written as:

$$W \cdot X + b = 0 \quad (2.12)$$

where W is the weight vector for optimal hyperplane, b is the bias, and $W \cdot X$ is dot product of weight and input vectors. Thus, we will consider a decision function of the form:

$$D(x) = \text{Sign}(W \cdot X + b) \quad (2.13)$$

The sign function is influenced by the sign of $W \cdot X + b$ and not its magnitude. In other words, the decision function is left invariant if we scale W and b by any positive quantity. Therefore, we can implicitly fix scale by fixing the canonical hyperplanes:

$$W \cdot X + b = -1 \quad (2.14)$$

$$W \cdot X + b = 1 \quad (2.15)$$

By subtracting the equation of one of the canonical hyperplanes from the other for the two support vectors on each side of the hyperplane ($X_1 - X_2$) we get:

$$(W \cdot X + b = -1) - (W \cdot X + b = 1) = W \cdot (X_1 - X_2) = 2 \quad (2.16)$$

the margin will be given by the projection of the vectors ($X_1 - X_2$) onto the normal vector to the separating hyperplane, i.e., $W/\|W\|$, as illustrated in Figure 2.5:

$$\text{Margin} = \frac{W}{\|W\|} \cdot (X_1 - X_2) \quad (2.17)$$

requirement at a cost proportional to the value of ξ_i . The formulation of the SVM optimization problem with slack variables is:

$$\phi(W) = 0.5(W \cdot W) + C \sum_i^l \xi_i \quad (2.21)$$

$$\text{Subject to the constraint: } y_i [(W \cdot X_i) + b] \geq 1 - \xi_i \quad (2.22)$$

The margin can be less than 1 for point x_i by setting $\xi_i > 0$, but then one pays a penalty of $C \xi_i$ in the minimization for having done that. $\sum_i^l \xi_i$ gives an upper bound on the number of training errors. Soft-margin SVM minimizes training error traded off against the margin. The parameter C is a regularization term, which provides a way to control over-fitting.

SVMs can be used for both linear and nonlinear data. It uses a nonlinear mapping to transform the original training data into a higher dimension, and a different kernel function can be used with SVM which is:

$$K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j) \quad (2.23)$$

The most common kernel functions are [32, 33]:

- The Polynomial kernel:

$$K(x_i, x_j) = (\gamma x_i^T x_j)^d, \gamma > 0 \quad (2.24)$$

- The Radial Basis Function (RBF) kernel:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \gamma > 0 \quad (2.25)$$

- The Sigmoid kernel:

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \quad (2.26)$$

Here, γ , r , and d are kernel parameters.

SVMs are effective on high dimensional data because the complexity of the trained classifier is characterized by the number of support vectors rather than the dimensionality of the data. The support vectors are the essential or critical training examples as they lie closest to the decision boundary. If all other training examples

are removed and the training is repeated, the same separating hyperplane would be found. The number of support vectors found can be used to compute an (upper) bound on the expected error rate of the SVMs classifier, which is independent of the data dimensionality. Thus, an SVM with a small number of support vectors can have good generalization, even when the dimensionality of the data is high [7, 28, 30, 34].

2.2.3.3 Naive Bayes classifier

Naive Bayes (NB) is a probabilistic classifier based on applying Baye's theorem and is commonly studied in ML [20]. The basic idea of the NB classifier is to estimate the probabilities of categories for a given document by using the joint probabilities of features and categories. The following provides a description for NB derivation.

Let D be a training set of documents and their associated categories $c_1, c_2, c_3, \dots, c_j$ and each document, $d_i \in D$, is represented by a set of features $F = (f_1, f_2, f_3, \dots, f_m)$. The probability, i.e., the maximal $P(c_j|d_i)$ then can be derived from Baye's theorem as follows:

$$P(c_j|d_i) = \frac{P(c_j)P(d_i|c_j)}{P(d_i)} \quad (2.27)$$

$P(c_j|d_i)$ is the probability of the category c_j for a given document d_i , $P(d_i)$ is equal for all categories so, it can be ignored [35]. The equation in (2.27) then can be rewritten as:

$$P(c_j|d_i) = P(c_j)P(d_i|c_j) \quad (2.28)$$

The naive part of Baye's theorem is the assumption that the features are conditionally independent, i.e., the conditional probability of a feature given a category is assumed to be independent of the conditional probabilities of other features given that category. As a result, the probability of category $P(c_j)$ can be rewritten as below:

$$P(c_j|d_i) = P(c_j) \prod_{k=1}^m P(f_k|c_j) \quad (2.29)$$

where m is the number of features (f_m) that represent document d_i . The NB classifier determines the category of a test document as the following:

$$V_{NB} = \operatorname{argmax}_{j \in C} \left[P(c_j) \prod_{k=1}^m P(f_k | c_j) \right] \quad (2.30)$$

Where C is a set of all possible target categories, the category with the highest probability is chosen as the target category for the document.

There are two different methods used for the NB classifier. The simple one is known as the Multi-Variate Bernoulli (MVB) model. This model only takes into account the presence or absence of a particular feature and the value of each feature is either 0 or 1 (binary model). As a result, it does not capture the number of occurrences of each feature. In addition, the probability of a document for a given category is calculated by multiplying the probability of all feature values, including the probability of features that do not occur in the document [7, 36].

The other model which is the more popular one is called the Multinomial Naive Bayes (MNB) model. This model captures the feature frequency information in documents as explained by Mitchell in [20], where a document is represented using a set of these features.

To use the MNB model, a dataset needs to be created for training the NB classifier where all documents that belong to the same category are concentrated to create a single large document and feature frequencies are used for training the classifier. The category probability of a document is calculated by multiplying the probabilities for each feature that occur in the document. These individual feature frequencies can be treated as an event, and a document can be described as a collection of feature events [7, 28, 29, 37].

The MNB model is more effective in a majority of cases when compared to the MVB model [36]. The MNB model captures feature frequency information in a document where MVB does not. A study by Schneider [38] argues that the superiority of MNB over MVB is not because of its ability to capture feature frequencies, and it should be attributed to the way the two models treat features that do not occur in a document.

To estimate the necessary probability parameters to classify a document, the NB classifier requires a small amount of training data comparing to other models [39].

This is due to the assumption that the features are independent. In addition, only the variances of the features for each category are needed to be determined and not the entire features. [7, 28, 29, 37, 40-42].

Even though the independence assumption is unrealistic, NB has been found effective for many practical applications, as a result of assigning the correct category based on the maximum probability, not the accurate probability.

In theory, Bayesian classifiers have a minimum error rate in comparison to all other classifiers [30]. However, in practice this is not always the case, owing to inaccuracies in the assumptions made for its use, such as class conditional independence, and the lack of available probability data [43]. The NB classifier also retains accuracy, and it assumes category conditional independence. In addition, existing dependencies among features cannot be modelled by this classifier [7, 28, 29, 37, 40-42].

2.2.3.4 Decision Tree classifiers

The Decision Tree (DT) is an ML classifier that takes the form of a tree where a collection of training instances are used to construct a classification tree. The decision tree consists of nodes and it is a directed tree. As a result, all nodes have exactly one incoming edge and one or more outgoing edges. A node that has no incoming edge is called the root node. The root node is placed in space based on the information values of selected features that split the training instances into two or more sub-spaces for each possible value. Each node in the tree contains the right amount of information that would be needed to classify new instances. These information values for features are calculated using different FS techniques such as IG (discussed in 2.2.3.1).

A node with outgoing edges is called an internal node. The training instances are divided into two or more sub-spaces by the internal nodes for each possible value. A node without outgoing edges is called a leaf node. Each leaf node is assigned to one category. To classify an instance using the DT, the classifier tests each node starting from the root node and goes down until a leaf node is reached. The category of the instance is indicated by the leaf node [29, 44]. The most well-known algorithm in the literature for building DT is the C4.5, which uses IG [45].

The DT characteristics have been found to fit a number of practical problems, for example classifying diseases based on medical cases, assessing the credit risk of loan applicants by their likelihood of defaulting on payments, equipment malfunctions by their cause, detecting advertisements on the web, and identifying spam emails [20, 44].

Take the example of a real-world problem predicting whether a loan applicant will repay their loan or not. An example of the training set is one that contains records of previous borrowers identified by ID number (1, 2, 3, 4, 5, 6, 7, 8, 9, and 10) as shown in Table 2.2. These records contain the personal information of each borrower such as “Home Owner”, “Marital Status”, and “Annual Income” which form the features set (F1, F2, and F3) [46].

Table 2.2 Training set for predicting borrowers who will default on loan payments

	F1	F2	F3	Category
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125k	No
2	No	Married	100k	No
3	No	Single	70k	No
4	Yes	Married	120k	No
5	No	Divorce	95k	Yes
6	No	Married	60k	No
7	Yes	Divorce	220k	No
8	No	Single	85k	Yes
9	No	Married	75k	No
10	No	Single	90k	Yes

These records have two categories “Yes” and “No” which indicate whether the borrower did not repay their loan and become a “Defaulted Borrower” or the borrower successfully repaid their loans and is not a “Defaulted Borrower”.

A DT classifier is constructed from the training examples in Table 2.2, in order to classify new test records. First, the best feature that divides the training examples to

different branches is chosen by calculating the information values for each feature using an FS techniques (e.g. IG technique). In the case of this example the F1 feature “Home Owner” represents the root node for DT, as shown in Figure 2.6.

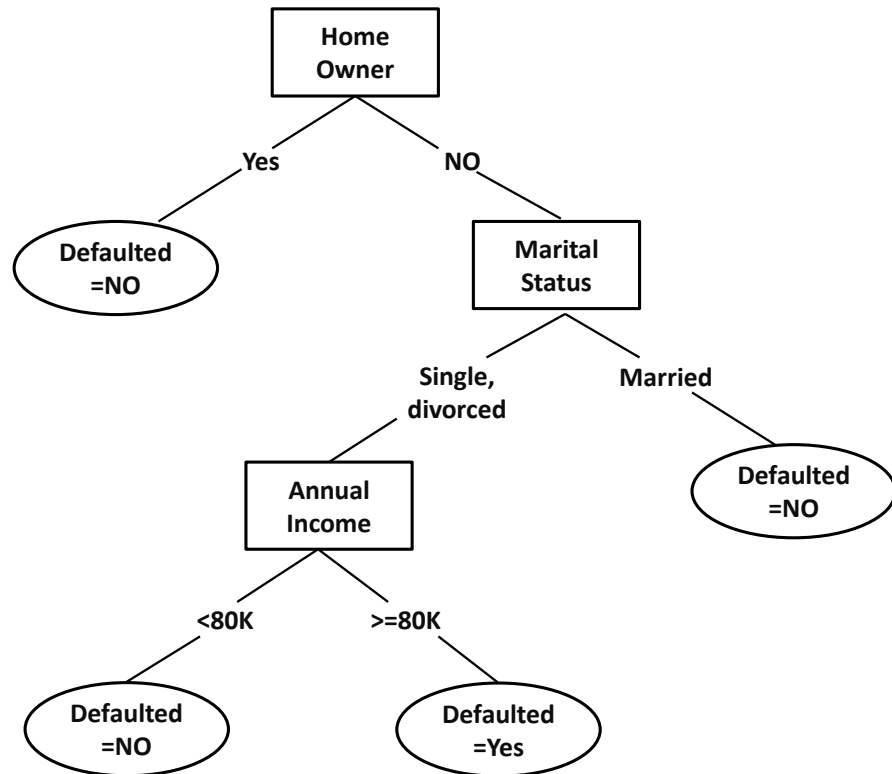


Figure 2.6 DT for constructed based on training examples

The root node F1 has two possible values: “Yes” and “No”. In the case of “Yes” where the borrowers are home owners, this leads to a leaf node with category “No” which means that they successfully repaid their loans. If the value of the case is “No”, meaning that the borrowers are not home owners, it leads to a child node with feature F2 “Marital Status”. The child node that contains feature F2 has three possible values: “Single”, “Divorced”, or “Married”. In the case where the borrowers are married, and they successfully repaid their loans, the value is “No” which is a leaf node. Both the value “Single” and “Divorce” lead to the next child node with feature F3 “Annual Income”. The value of feature F3 leads to two leaf nodes; in the case where the annual income of the borrowers is less than 80K, it will lead to the category of “No”, if the income is over 80K the value is “Yes”.

For example, to classify the borrower record (ID 11) shown in Table 2.3, using the DT classifier shown in Figure 2.6, The DT classifier is going to compare feature

values in the test example with the values of each node in the tree to choose the right branches until it reaches a leaf node that has the classification category as shown in Figure 2.7. The bold red line in Figure 2.7 shows how the test example is categorized using the DT classifier.

Table 2.3 Test example for an applicant with personal information to predict the category of the borrower

	F1	F2	F3	Category
ID	Home owner	Marital status	Annual income	Defaulted borrower
11	NO	Single	125k	?

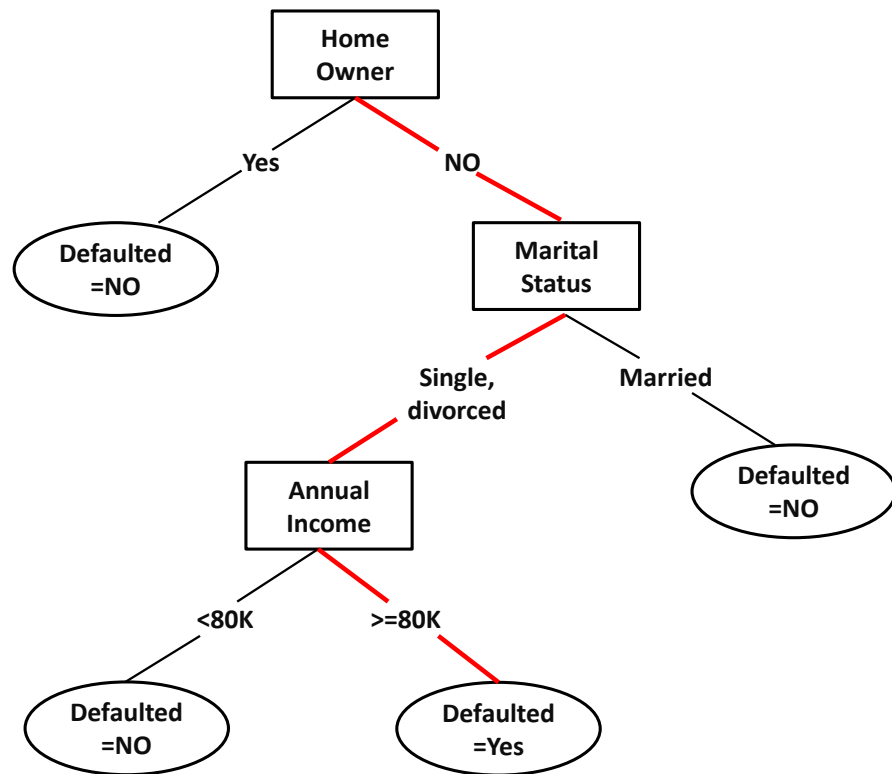


Figure 2.7 Illustration of how the DT classifies a test example

One of the advantages of DT is that the representation of the model is self-explanatory and easy to understand. It is clear why a classified instance belongs to a specific category [47].

The DT uses the divide-and-conquer method to divide the training space for building the classification tree. Even though this method is quick, efficiency can

become an important element when hundreds of thousands of training instances are involved. The most time consuming aspect is sorting the instances on a feature to find the best split. In addition, decision trees perform more efficiently when a few relevant features are involved, but they are less proficient when many complex relevant features are present. DT is over-sensitive to the training set and noise [45, 47].

The DT suffers from the problem of over-fitting which happens when the ML algorithm continues to develop hypotheses that reduce training set errors at the cost of an increased test set error [20]. The issue of over-fitting in DT could be avoided by pruning the tree after its full growth or preventing it from reaching its maximum size and over-fitting the training examples [20, 44].

2.2.3.5 Random Forest classifiers

Random Forest (RF) is a supervised ML algorithm commonly used in classification and regression problems. RF was proposed by Breiman in 2001 [48] based on ensemble learning, where a set of classification trees are built based on the bagging method. To grow each tree, a random sample is selected, with replacement, from the original training set to create a new one of the same size [49]. For each node, the RF classifier uses a random selection method to select a subset of features [50] and the best split is selected based on the information value of these subset features. After a large number of trees are generated, the most popular category is voted for by these trees.

Let D be a training set of documents and their associated categories, $D = \{(\vec{d}_1, c_1), (\vec{d}_2, c_2), (\vec{d}_3, c_3), \dots, (\vec{d}_i, c_j)\}$, where i is the number of documents (examples) in the training subset. The $(\vec{d}_i, c_j)$, is a pair of variables such that $\vec{d}_i$ is a feature vector of document $d_i \in D$ and c_j is a category $c_j \in C$.

The RF classifier takes the original training set and grows several different training subsets from it, each training subset builds a tree, and combines them by voting to predict the category for a test instance. For example, for the training set D , the RF classifier creates several different training subsets using bagging such as $D_1, D_2, D_3, \dots, D_B$, where B is the number of training subsets and each one will be used to build a tree such as $T_1, T_2, T_3, \dots, T_B$. These training subsets are not identical but are

certainly not independent where instances are randomly sampled, with replacement, from the original dataset to create a new one of the same size. Using sampling with replacement ensures that selected instances are not removed from the training set, and procedure inescapably replicates some of the instances and deletes others [7].

As a result, trees based on these training subsets do not make the same prediction for each new test instance. The small changes in the input data can lead to quite different classifiers when the learning algorithm is unstable such as in trees; in this case, bagging helps by neutralizing the instability of learning algorithms. This is done by combining these multiple classifiers which generally decreases the expected error by reducing the variance component.

Bagging introducing randomness into the input data, but there are other ways of creating diversity by introducing randomization. The randomizing variable θ uses different, randomly chosen subsets of features. For each node S_j in tree $T_b \in T_B$, a random variable θ_k is generated, independent of the past random vectors $\theta_1 \dots \theta_{k-1}$ but with the same distribution [48]. This variable θ is used to determine the selection of the coordinate to split and the position of the split for the data using the equation in (2.31) that quantifies the entropy of the node S_j using IG.

$$I_j = H(S_j) - \sum_{i \in \{L, R\}} \frac{|S_j^i|}{|S_j|} H(S_j^i) \quad (2.31)$$

where,

- $H(S_j)$ is the entropy of the node S_j for all the dataset D_B .
- $H(S_j^i)$ is the entropy of the children of node S_j .
- $|S_j^i|$ is the portion of the dataset D_B for each child of the node S_j which could be left or right.

After calculating all random variables $\theta_1, \dots, \theta_{k-1}, \theta_k$ for the node S_j , the best θ_j is selected using the equation in (2.32)

$$\theta_j = \text{avg max } I_j \quad (2.32)$$

Each node is working as a classifier $h(d_i, \theta_j) \in \{L, R\}$, where d is an input vector of data in D_b and the output from the classifier function is splitting the input vector d to the right or left child node based on the random variables θ_j . These classifier

functions are kept in the tree nodes to be reused to classify test instance. The same steps are repeated for each node until the tree T_b is grown. The same steps are repeated until all trees $T_b \in T_B$ are grown. To classify a test instance, each tree $T_b \in T_B$ is used to produce a category, and the most popular category is voted for.

In the model we have in mind, the variable Θ is assumed to be independent of $\vec{d}_i$ and the training D_n . This excludes in particular any bootstrapping or resampling steps in the training set. This also rules out any data-dependent strategy to build the trees, such as searching for optimal splits by optimizing some criterion on the actual observations. However, it allows Θ to be based on a second sample, independent of, but distributed as D_n . [48]

An RF classifier requires less time than other classification algorithms to implement as a result of its simplicity. In addition, the classification predictions are also accurate, and the RF can handle a large number of input features and large datasets.

2.3 Performance Evaluation

Evaluating the performance of ATC systems is important as it shows their effectiveness and ability to make correct classification decisions. In addition, it allows for comparing different components of ATC systems and their impact on the overall classification performance.

Practically, the performance of an ATC system can only be evaluated experimentally using a set of pre-classified (i.e., categorised) documents. In developing the system, this set of documents is divided into training subsets which are then used to build a classifier model and a testing subset to assess the accuracy of classification using a given performance measure. For estimating the accuracy of the obtained classifier, two common approaches are usually used to assess the ability of the classifier to predict the correct category of unseen instances, the hold-out, and cross-validation.

2.3.1 Hold-Out

In the hold-out method, the dataset is arbitrarily split into two separate subsets, a training subset and a test subset [51]. Usually two-thirds of the dataset is retained

for training, and the remaining third is used for testing. The training subset is used to learn the classification algorithms and to build a classifier that predicts the categories of a testing subset. A problem may arise when one of the categories is not represented in the training portion of the data [7, 52]. This problem is solved using what is called a stratified hold-out. In this case, the selected sample contains instances from all categories of the dataset, and all categories are represented in both training and testing subsets of the dataset. Furthermore, the hold-out approach is a pessimistic estimate because only a portion of the data is given to the classifier for training and the accuracy increases as more instances are seen and as a result a smaller test set means the confidence for the accuracy will be wider. On the other hand, the more instances left for the test set will lead to a higher discrimination for the classifier [53].

2.3.2 *K*-fold Cross-Validation

In this method, the dataset is randomly split into K equal subsets or folds. Each time, a different part of the fold is used as the test subset and the other remaining for training subsets. After finishing training and testing all folds, the results for K -folds are averaged. This approach is commonly used to smooth out the variations in the dataset. In addition, a stratified version of the K -fold cross-validation is used to ensure that all given categories are represented in all folds and each fold is a good representation of the whole dataset in a way that if a category is about 20% of the dataset size it also will be about 20% of the fold size [7, 20, 52].

A classifier is considered stable for a given dataset, if it makes the same predictions when the dataset is disturbed by some changes. By increasing the size of distributions, stability for the classifier is less likely to hold. The stability of a classifier holds more in a 20 fold cross validation than in a 10 fold cross validation and both should be more stable than a hold-out of one-third. The classifier is unlikely to be stable for large distributions unless it has reached its maximum learning capacity. Using 10 fold stratified cross validation is the best method for model selection in TC experiments.

2.3.3 Performance Measures

To evaluate the performance of ATC systems, different measures are used such as accuracy, Micro and Macro F1 measures. To compute these measures, four values are needed. These values are generated after a classifier is used to predict the category of a test document: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) as can be seen in Table 2.4.

Table 2.4 Classifier possible predictions in an ATC system

Actual	predicted	
	Yes	No
Yes	<i>TP</i>	<i>FN</i>
No	<i>FP</i>	<i>TN</i>

“Positive” means that the document was classified as belonging to the category, whereas “negative” means that it was not. “True” means that the classification was correct and “false” that it was not, such that:

- TP: refers to the cases when both the classifier and human cataloguer agree on assigning a category to document d ;
- TN: refers to the cases when both the classifier and human cataloguer agree on not assigning a category to document d ;
- FP: refer to the cases when the classifier has mistakenly (as judged by a human cataloguer) assigned a category to document d ;
- FN: refers to the cases when the classifier has failed (as judged by a human cataloguer) to assign a correct category to document d .

Accuracy is the percentage of documents that are classified correctly. It is computed by dividing the number of documents correctly assigned to category c_j by the total number of documents in the testing dataset [54, 55].

Using the four values defined above, *accuracy* is defined as:

$$A(c_j) = \frac{TP_j + TN_j}{TP_j + TN_j + FP_j + FN_j} \quad (2.33)$$

From the four values, we can define the standard evaluation metrics precision and recall to compute the F1 measure. *Precision* is the proportion of predicted category c_j for a document that was correctly predicted. *Precision* (c_j) is defined as:

$$P(c_j) = \frac{TP_j}{TP_j + FP_j} \quad (2.34)$$

The *recall* is the proportion of actual category c_j for a document that was correctly predicted. *Recall* (c_j) is defined as:

$$R(c_j) = \frac{TP_j}{TP_j + FN_j} \quad (2.35)$$

The F1-measure is the harmonic mean of the two:

$$F1(c_j) = \frac{2P(c_j)R(c_j)}{P(c_j) + R(c_j)} \quad (2.36)$$

These measures are relevant to single category classification problems, i.e., when an example may be either positive or negative. To use precision, recall, and F1 measure values for the various categories, these measures should be combined to obtain an accurate measure of the ATC algorithms. There are two possible methods that can be used: Macro averaging and Micro averaging.

Macro-averaged performance scores are calculated by first computing the scores for each category and then averaging these category scores to compute the global means where Micro-averaged performance scores are computed globally over all categories. There is an important distinction between Macro-averaging and Micro-averaging. The Micro-averaged performance score operates at the document level and gives equal weight to every document. In addition, it is primarily affected by the categorization accuracy of the larger categories. Therefore, it is considered a per-document average. On the other hand, the Macro-averaged performance score averages results over all categories. As a result, it gives equal weight to every category, regardless of its frequency, and is therefore a per-category average.

2.4 Summary

In this chapter, the Automatic Text Classification (ATC) systems are discussed. The main goal of ATC is to eliminate the need for manual organisation of large collections of text. To achieve that, a number of steps are involved to classify a document automatically. In addition, the key components to build an ATC system were presented, namely text pre-processing, text representation, and classification models. For the text pre-processing component, the main steps that involve converting text to a well-defined set of features are discussed. These features are then used in the text representation component where each document is represented as a set of weighted features in a Vector Space Model (VSM) and then fed into ML based classification algorithms. Moreover, the most popular and successful ML algorithms which are frequently applied to TC problems were discussed, namely Support Vector Machines, Naive Bayes, Decision Trees, and Random Forest. In addition, Feature Selection (FS) techniques were also discussed.

Finally, to evaluate the performance of an ATC system and to show the system's effectiveness and its ability to make correct classification decisions, different performance measures are presented. In addition, different methods that divided a set of documents into training and testing subsets were reviewed.

CHAPTER 3

Text Representation

3.1 Introduction

Text representation is an important aspect of a successful ATC system. In this chapter, different types of features used in text representation are discussed. As previously mentioned in Section 2.2.2, the text is traditionally represented using the Vector Space Model (VSM) which is a weighted high dimensional vector, where each dimension corresponds to a unique feature; these features can be single words, N-grams, or phrases. Features can also be concepts from a Knowledge Base (KB) such as WordNet or Wikipedia, or a combination of both concepts and words that use different strategies to build combined text representation models.

3.2 Bag of Words (BOW) Representation

The most popular representation model used in Information Retrieval (IR) is the Bag of Words (BOW) which owes its popularity to its simplicity and efficiency. In this model, the text is represented by words, each word being an independent feature on its own [56].

Each dimension in the VSM corresponds to an individual word, and if a given word occurs in the text, based on the type of BOW used, the value of the word could be defined as a binary or frequency vector. In the case of a binary vector, if the word appears in the text the corresponding value is one, otherwise, if the word does not appear in the text, the value is zero. In the case of a frequency vector, if the word appears in the text the corresponding value is the number of times the word appears in the text, otherwise, if the word does not appear in the text, the value is zero. As described in Section 2.2.2, there are different weighting schemes used in IR and text mining applications.

BOW has been used in many text mining tasks including ATC [25], text clustering [57], sentiment analysis [58], and document summarization [59]. Various

researchers have shown the BOW's efficiency not only in English ATC but also in other languages such as Arabic [60], Portuguese and Spanish [61], Japanese [62], and Chinese [63].

However, despite its popularity and efficiency, the BOW has a number of limitations:

- Issues with synonymy, that is, problems with words with equivalent meanings. The BOW model treats synonymous words as independent features, e.g., “classification” and “categorization” are considered as two independent words with no semantic association. As a result, with the BOW documents that discuss similar topics and contain synonymous words can be considered as unrelated.
- Words can have different meanings depending on their surrounding context, i.e. polysemy. The BOW model cannot discern the meaning of a single word in different contexts even if the meaning is totally different. Take the word “bright” for example, depending on its context “bright” could mean shining clever, vivid or cheerful. Using the BOW as a representation model the word “bright” will be treated as a single feature irrespective of its intended meaning in different contexts.
- The BOW model representation breaks terms into their constituent words, e.g., it breaks “text classification” into the words “text” and “classification” without considering the order of the words in the text or the semantic connections between them. The semantic connection does not cover only synonymy and polysemy, it also reflects the relationship between words, for example, “text classification” is related to “text mining”.

Due to these limitations, researchers have attempted to address the above issues using other features in the text representation model, such as N-grams.

3.3 N-gram based Representation

In general, an N-gram is a contiguous sequence of N items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words, or base pairs according to the application [2]. In this section, an N-gram is defined as a sequence of N consecutive characters that appear in a longer string. An N-gram

can be accompanied by any set of characters in a string; N could be equal to 1, and the N -gram is a letter or 2-grams known as bigrams and 3-grams are called trigrams [64]. For example, in the case of $N=3$, the string “text classification” yields the following set of 3-grams: “tex”, “t c”, “cla”, “ssi”, “iff”, “cat”, “ati”, “ion”.

N -grams were first mentioned in the context of the theory of communication by Shannon in 1948 [65]. Since then, N -grams have been widely used in IR applications [66] and ATC [2, 64].

Each word in various human languages occurs with a different frequency, and some words occur more frequently than others. This idea has become known as Zipf's law where “the n th most common word in a human language text occurs with a frequency inversely proportional to n ” [67]. Zipf's law concludes that there is always a set of words which have the most frequency of use in the language compared to other words. This is true for both general words and words that are specific to a particular subject. Zipf's law is also applicable to the frequency of N -grams. This suggests that documents following the same category will have comparable N -gram frequency distributions [2].

N -gram based representation has the capability of dealing with noise in text such as mistakes caused by human error (spelling for example) or errors generated during the Optical Character Recognition (OCR) process [2]. For example, in a BOW model sample the words “colour” and “color” are considered as two distinct features, but in an N -gram model, a set of feature values which are intersect is created by these words, and therefore a part of the association between the words are retained.

N -gram based representation is language independent and can be applied to texts written in languages with no spaces to separate the words such as Japanese and Chinese [68, 69]. Therefore, a text which is represented using the N -gram does not need to use the stop words removal or morphological normalization [66, 70]. As a result, using N -grams as features in the representation model for classification problems in other languages such as Arabic [71], Croatian [72], Japanese [73], and Chinese [69] shows efficiency even though their morphological problems are more complicated than English.

Many researchers [66, 72] note that in English text, when using N-gram based representation with the length of four ($N=4$), the total number of unique N-grams converges to the number of unique words found in a large dataset, where if the length of the N-grams is five or more, the total number of unique N-grams is larger than the total number of words found in the text which leads to high-dimensional feature in VSM [66, 72].

Unfortunately, N-gram based representation has a number of shortcomings:

- In large datasets, extracting N-grams will yield a large number of possible N-grams which leads to the problem of extremely high dimension vector representation for documents. As a result, the document representation vectors consume a large amount of memory which slows down the classifier learning process and the classification of new documents [72].
- N-gram based representation acts like BOW-REpresentation when dealing with synonymous words by treating them as independent features. In addition, it generates completely different sets of N-gram frequency profiles for documents containing synonymous words. As a result, a large distance is calculated between the frequency profiles of the document and the category leading to misclassification.
- N-gram based representation does not reflect any semantic connection between words in regards to the order of their appearance in a text or the different meanings that words could have depending on their surrounding context as it is a sequence of N characters that could appear anywhere in a longer string.

Due to these shortcomings, a feature which is less ambiguous than N-grams and more specific than words has to be used, such as phrases.

3.4 Phrase based Representation

Phrases are defined as sequences of words that appear in a text. They provide more specific meaning compared to single words and less ambiguous. For example, classification is a general term that acquires a special meaning by specifying the object such as text classification, image classification, query classification, and question classification or single- and multi-label classification.

Two documents that mention the same phrase such as “question classification” are more similar than those that only share the word classification. In addition, phrases help to solve the ambiguity problem.

In general, two different types of phrases have been proposed and investigated as features in the text representation model for the ATC system:

- Syntactic phrases. These types of phrases make use of lexical and syntactic rules of phrases to extract the terminologies in the text such as noun phrases and verb phrases and uses these linguistic units as features in the representation model.
- Statistical phrases. This type of phrase makes use of the sequence of N words occurring consecutively in the text. It ignores the semantics in texts and builds phrases after removing stop words. Phrases are either words grouped based on co-occurrence or a word sequence extracted from documents by the traditional string matching method. In many studies, authors refer to statistical phrases as N -gram where N is the number of words in the phrase.

Both syntactic and statistical phrases have been studied in ATC and IR applications. For example, the first work that studied the use of syntactic phrases in ATC was done by Lewis [74, 75]. In this study, using phrases as features and NB as the classification algorithm yields significantly lower performances than the standard BOW model. In Lewis’ work, only syntactic phrases are used to represent the text where in most other works phrases are added to the BOW model. Another study [76] implemented a practical method to extract syntactic phrases from documents. In addition, two strategies were used to represent documents using the extracted syntactic phrases, the general and sub-topic representation. In the general representation only short syntactic phrases that point out general concepts in documents are selected, whereas in sub-topic representation only long syntactic phrases that refer to more specific concepts in documents are selected. A Feature Selection (FS) technique, Information Gain (IG), was used to remove phrases with low information value. The SVM classification algorithm was used to classify the Reuters-21578 documents. The results demonstrated that representing documents with syntactical phrases using sub-topic representation outperforms the general one. Furthermore, the representation using BOW without linguistic pre-processing outperforms any

representation using syntactical phrases with any other strategy [77]. This is consistent with other works such as in [27] which reports the same result on Reuters-21578.

The reason behind this poor result is that unless phrases occur frequently enough in the document collection, they are unlikely to make an impact in terms of effectiveness of the classification system. Another reason is that not every syntactic phrase denotes an interesting concept and distinguishing interesting phrases is difficult [78]. Therefore, a number of researchers used statistical phrases in ATC systems to improve the low performance of syntactic phrases.

Statistical phrases have a number of advantages over syntactic ones by means of stronger and less computationally difficult algorithms. In addition, the effect of unrelated syntactic phrases can be removed, and worthless phrases tend to be filtered out from useful ones [76]. For instance, the study by Mladenic [79] extracted statistical phrases of a length of up to five using an extraction algorithm that relies on document frequency as a statistical filter. For the classification algorithm, they used NB on a body of web pages. They report that statistical phrases of a length of up to four gave significant benefits with respect to the BOW, while statistical phrases of a length of five did not provide additional benefit. In [80] authors used an algorithm similar to the one used in [79] for extracting statistical phrases of a length up to 5. The Reuters-21578 dataset is used in the experiment and statistical phrases of a length of 2 show significant improvement in performance. Another finding concludes that the longer the lengths of the statistical phrases, the classification performance is reduced. The author used a dataset of Usenet newsgroups articles and reported that statistical phrases of a length of 3 are of some use, whereas the negative contribution of N statistical phrases was confirmed. The authors in [76] defined the statistical phrase as a stemmed and alphabetically ordered sequence of N words. Different lengths of statistical phrases were extracted and used to represent documents, and different FS techniques were used to remove phrases with low information value. The Racchio algorithm was used to classify the Reuters-21578 dataset, and the experimental results showed that using statistical phrases as features to represent documents in the TC task did not yield better performance than the BOW. Other researchers combined words and statistical phrases for text representation to classify text documents such as [81]. In comparison with using the BOW

representation, using both BOW and phrases for text representation did not show significant improvement in terms of classification accuracy.

Unfortunately, both statistical and syntactic phrases as representation features for text fail to be a satisfactory solution to the problems in the BOW model. For that, a new feature is introduced in text mining to represent text which is concepts.

3.5 Bag of Concepts (BOC) Representation

Concepts as features to represent text are less ambiguous than N-grams and more specific than words and overcome the low performance of phrases. A concept is a unit of knowledge providing a unique meaning. Synonymous words are mapped to the same concept which provides that unique meaning that they share. Words with multiple meanings are mapped to different concepts corresponding to their intended meanings based on the surrounding text. For example, “classification” and “categorization” are recognized as one feature, while texts about programming language and the island of Indonesia will not be mistakenly connected if both of them mention “Java”. Using concepts to represent the text reduces the complexity and dimensions of VSM. As a result, the number of identified concepts in a text is usually far less than the number of words, where concepts are efficient built-in representations which mapped synonymous words to one entity.

Building a BOW representation is relatively easy (see Section 3.2) in comparison to building a BOC representation as the BOC needs an external concept system (knowledge base) to provide the concepts. To build a BOC representation from a text, concepts from an external Knowledge Base (KB) are nominated and used as features to represent the text. The KBs can be divided according to their characteristics such as the type of concepts, how concepts are structured, organization, coverage, and extensiveness.

3.5.1 Knowledge Bases

Knowledge Bases (KB) can be encyclopedias or lexical resources depending on the type of KB they are. Examples of suitable encyclopedias that provides concepts for text representation are the Encyclopedia Britannica [82] and Wikipedia [83].

The Encyclopedia Britannica and Wikipedia are considered as comprehensive knowledge bases. In terms of size, Wikipedia is larger as it has over 4 million articles (concepts) whereas Encyclopedia Britannica contains only 100,000 articles. In addition, Wikipedia is a freely accessible knowledge base whereas Encyclopedia Britannica is not.

A study by Giles [84] established that Wikipedia's accuracy rivals that of Britannica. Therefore, Wikipedia as a knowledge base has been widely and successfully used to assist better comprehension of text. Many studies have used Wikipedia concepts for representing text, for example, in text categorization [85, 86], for IR applications [87, 88], and text clustering [89, 90]. In these studies, concepts which are Wikipedia articles become features that are used to represent text, and to recognize these concepts, words and phrases in these texts are mapped to Wikipedia articles that match their intended meaning.

Lexical resources have also been used as knowledge bases to provide concepts. These types of knowledge base usually give information about individual words rather than general conceptual knowledge [91]. Examples of such lexical resources include WordNet and Roget's thesaurus [92], where WordNet is widely investigated to provide concepts for text, and Roget's thesaurus is less explored. These resources have different characteristics, for example, with WordNet each concept has a description and a list of synonyms associated with the meaning of the concept. WordNet also has many relations between concepts (see Section 3.4). On the other hand, in Roget's thesaurus, concepts are not synonyms but can be observed as connotations of the meaning of a concept as it encodes lexical knowledge and web directories to provide concepts. Roget's thesaurus suffers from limited coverage as it only has 1000 concepts, which is small compared to WordNet's 206,941 concepts. Therefore, many studies use WordNet concepts for representing text in text categorization [93, 94], IR applications [95, 96], measuring semantic relatedness between texts [97, 98], and text clustering [99, 100].

Lexical resources usually cover common words and expressions which do not have corresponding concepts in encyclopedias. For example, about 19% of WordNet concepts are adjectives and adverbs [101] which are not represented in encyclopedias.

Another resource that has larger coverage than other knowledge bases is the Open Directory Project (ODP) [102]. Even though ODP has hundreds of thousands of concepts, it still remains a less explored resource. This is because ODP suffers from insufficient semantic information. ODP does not capture the rich semantic relations between concepts as WordNet and Wikipedia do.

Another way to divide KBs is on domain generality, where they can be divided into domain independent or dependent resources. Domain independent resources such as Wikipedia, ODP, and WordNet can use their concepts to represent text that cover general topics from different domains. Domain dependent resources cover concepts in specific domains. The research and applications of domain dependent are usually restricted to processing text from a single domain [103, 104] such as the Medical Subject Headings (MeSH) [105] and Agrovoc [106]. These KBs organise concepts based on their hierarchical relations where superordinate concepts are more generic and broader than subordinate ones.

In the context of an ATC system, domain dependent KBs are more powerful when texts match the KB's domain which is confirmed in [104]. The study showed that the MeSH thesaurus concepts give better results than WordNet concepts when used with the OHSUMed collection [107] because both share the medical domain.

In addition to the previously made distinctions, KBs can also be categorized as controlled vocabularies, glossaries, thesauri, and ontologies based on the comprehensiveness of the semantic information encoded in them [108].

A knowledge base can be classified as a controlled vocabulary when a list of terms is built manually for each concept. The main reason for using the controlled vocabulary is to ensure that a concept is always referred to using the same term. Controlled vocabularies reduce ambiguity and divergence, and it is used in many applications such as cataloguing and metadata assignment.

On the other hand, when vocabulary is providing a definition of each term, the knowledge base is classified as a glossary. However, glossaries can be automatically created and maintained where the controlled vocabulary is manually built, and vocabulary is not necessarily controlled [109].

When the KB encodes relations between concepts such as synonyms, broader and narrower terms, it can be classified as a thesaurus such as the Library of Congress

Subject Headings (LCSH), MeSH, and Agrovoc. Ontologies extend thesauri by introducing more strictly defined relations such as inheritance and properties.

By choosing the appropriate KB, based on its characteristics and how it maps words in the text to identify concepts, it provides more effective concepts to be used in text representation.

3.5.2 WordNet as a Knowledge Base

WordNet is a large lexical ontology where a concept in WordNet is expressed by grouping nouns, verbs, adjectives, and adverbs into sets of cognitive synonyms [110]. Concepts in WordNet group words together based on their meanings. The main relation between words in WordNet is synonymy. A concept carries a unique meaning in WordNet, so words that carry several distinct meanings are represented with many distinct concepts. These words are polysemous where words with only one meaning are monosemous.

Additionally, a concept contains a brief definition that explains the concept and clarifies the use of other concept synonyms. Concepts are interconnected by semantic and lexical relationships which make them useful for linguistics and NLP. The most frequent relationship between concepts is the super-subordinate relation (hyponymy). The hyponymy relationship links more general concepts to specific ones, for example, from “accommodation” to “house”. The hyponymy relationship is transitive where if a semi-detached house is a type of house, and if a house is a type of accommodation, then a semi-detached house is a type of accommodation. Another relationship is meronymy, which is the part-whole relationship held between concepts like house and backyard and second floor. Parts are inherited from their superordinate; if a house has a backyard, then a semi-detached house has a backyard as well. Parts are not inherited “upward” as they may be characteristic only of specific types of things rather than the class as a whole, for example, different types of accommodation have backyards, but not all types of accommodation have backyards such as apartments.

To use WordNet concepts to represent text, two steps are involved. First, nominee concepts in the text are distinguished, and then select the most probable concepts for polysemous words by disambiguating words based on their actual meanings in the text.

To identify WordNet concepts in the text, first all different forms of words such as inflectional variations, pluralities, and tenses are mapped to their corresponding base forms and bring together morphological variations for words. WordNet as a KB provides a tool with built-in functions that mapped different form of a word to its base form. The reason behind that is the actual meaning of the word is changed with these derivational variations, and therefore it is important to keep totally different concepts for both the base word and variations forms of the word. For example, “sad” and “saddens” are recognised as distinct WordNet concepts. In addition, concepts in WordNet have lexical categories which are one of the four Parts of Speech (POS), namely nouns, verbs, adjectives and adverbs. By identifying the POS categories of concepts, the accuracy and efficiency of candidate selection and the sense disambiguation task is improved. For example, the word “mean” corresponds to 15 WordNet concepts which are classified based on their POS to three categories (nouns, verbs, and adjectives). By knowing the POS categories of concept, only those with the right POS category are used in the sense disambiguation algorithm.

The goal of the sense disambiguation task is to determine the intended meaning of polysemous terms and as a result, reduce the level of noise and loss of information in text representation. WordNet uses two rules for sense disambiguation, which are the first common sense rule and the based-on-context sense rule. The simplest method is the first common sense rule which chooses the first concept in the ranked list. Concepts in WordNet that correspond to an ambiguous word are ranked in descending order based on their commonness, where the most common are placed first and the least common are placed last. The order of concepts is derived from word frequencies obtained from a large standard English corpus such as the British National Corpus [111]. This method has been effective in a number of studies such as in [99, 112]. By using the most common sense for ambiguous words, the disambiguator always chooses the most likely sense of a word irrespective of its surrounding context. Therefore, the need to disambiguate words based on context is appealed and a rule based on context is introduced. In this rule, each concept in the ranked list that corresponds to an ambiguous word is checked. The concept with the best fit with the surrounding context is selected. This rule leverages unambiguous words in the context to find the most probable sense for the ambiguous words. For example, the word “Java” is ambiguous and three

different concepts correspond to it such as “coffee”, “Island of Indonesia” or “Java programming language”. Using the context based disambiguation method, if the word “Java” occurs with words like “cup”, “beverage”, and “beans” which are unambiguous, it will refer to coffee whereas if it occurs with words like “programming” or “object-oriented programming” it means Java as a programming language. It is more likely that the selected concept is the correct one when the degree of the relationship between a concept’s meaning and unambiguous concepts in the text is higher and as a result, it fits the context better.

3.5.3 Wikipedia as a KB

Wikipedia is the largest electronic knowledge repository on the Internet, and its content is entirely contributed collaboratively by volunteers. Wikipedia is a comprehensive, up-to-date, and a well-formed knowledge source. Each Wikipedia page is unique and has a title and type (article, category, or redirect). Each article in Wikipedia describes a single concept, and the article title is the descriptor of the concept. The title is brief, well-formed and can be used as a descriptor in ontologies. The aim of the redirect pages in Wikipedia is to connect articles with alternative titles that correspond to their synonyms, acronyms, and common variant spellings. For example, the Wikipedia article Java (programming language) describes the history of general-purpose computer programming language, the basic practices of the Java, describes the implementation, an example of Java syntax. It also has 31 redirects, including synonyms like a Java computer language and Java (computer), morphological variations like JAVA, and even common spelling mistakes like “Java (langage)” and “Java (language)”. In Wikipedia, an article may have more than one redirect and other Wikipedia articles may link to it. Each link to an article has an anchor text. Anchor texts are short phrases that articles use when referring to other articles. They provide a large number of alternative expressions for the article being pointed to, such as synonyms and other variations in surface form. For example, 246 anchors found in Java (programming language).

The great majority of Wikipedia articles, i.e., concepts, are classified to one or more Wikipedia categories, which are hierarchical and descend from a single root. A concept can belong to multiple parent categories which are more general articles

than the concept itself. In addition, more specific concepts which are child articles can be mined by the concept as a parent category for them. All these elements offer additional information, and they are grouped into labels. These labels will be used as concepts to represent a text in different applications.

To use Wikipedia concepts to represent text, two steps are involved which include detecting candidate concepts in text and disambiguating concepts based on the intended meaning. Several methods have been used to address each step as the representation model effectiveness is dependent on these steps. To identify Wikipedia concepts in a text, the first method involves only string matching between the titles of Wikipedia articles and words in the text. This method is strict where only articles with a positive match are linked to the text. However, this method is limited to article titles and therefore variations in expression are ignored. [113]. Another method proposed by Gabrilovich and Markovitch [85] select Wikipedia articles as a candidate concepts by analysis and matching between the text and Wikipedia articles content of the text. This is less limited than the string matching method, but it is computationally costly because it involves full text level analysis to measure the relevance between a Wikipedia article and a given text. The method indexes the text with all the concepts in Wikipedia resulting in a complex VSM with millions of features. In studies [114, 115] researchers used another factor to connect words in running text to Wikipedia articles which is the anchor text in the Wikipedia pages. By matching the text with the anchor texts that refer to the concept, the Wikipedia concept is identified in a free text. Using this method to identify concepts in the text is more flexible and efficient than the text overlap method and strict title matching method where a full text parsing and matching are avoided. In addition, it avoids what the text overlap method is doing by indexing a text with millions of Wikipedia concepts.

After selecting Wikipedia concepts from a text, a sense disambiguation step is required for ambiguous concepts. The sense disambiguation is controlled in an unsupervised way in WordNet, whereas in Wikipedia it is supervised. Supervised methods usually are more effective and accurate than unsupervised ones where labelled data is needed to learn how to disambiguate anchor text. The huge number of anchors in Wikipedia provides a large amount of labelled data. Where a positive example for context created by every inter-article link in Wikipedia that specifies the intended sense of the anchor, and negative examples are provided all

other possible senses for the anchor other articles that it has been used to link to. Both studies in [114, 115] used supervised machine learning techniques to perform disambiguation with Wikipedia concepts and confirmed that they outperform unsupervised heuristics.

The association between an anchor phrase, its surrounding context, and its target article are characterized by designing different features such as in [114, 115]. The study in [114] used the ambiguous word itself with its POS, directly neighbouring words and their POS, and most five words that are at least appearing three times in the contexts which defining a certain unambiguous word sense as features. Where authors in [115] used different sets of numeric features such as the prior probability of sense being the intended one for the given phrase, the average relatedness of the sense to the phrase's surrounding context, and the overall quality of context. The context is represented by the set of unambiguous anchor phrases in it, which only points to a single article throughout Wikipedia.

To disambiguate a concept in free text, features are calculated for each sense of the concept and a classifier such as NB or DT is trained with examples from Wikipedia and each example is represented by these features. The trained classifier predicts the most likely target concept. Even though Milne and Witten, the authors in the study [115], avoid NLP and used less features compared to what was used in [114] their method achieves competitive performance and is more efficient.

3.5.4 A Review of the BOC with Wikipedia/ WordNet

WordNet has been used as a KB since 1997 when Rodríguez [116] enhanced Rocchio and neural network learning algorithms with WordNet concepts to improve the representation of categories. The proposed enhancement significantly improves the classification accuracy on the Reuters-21578 dataset. Based on the assumption that the name of a category can be a good predictor of its occurrence, the author performs a category expansion in a similar way to query expansion in IR. For any category, the WordNet concepts that represent the meaning of the category are selected, and any word belonging to them is added to the word set to represent the category. In the study, the disambiguation process to select the candidate concepts is performed manually, as the small number of categories in

the test collection made it feasible. The results of the study showed that combining words and WordNet concepts performs better than using only words to represent categories by 20%. In addition, combining words and WordNet concepts improved the classification accuracy of documents which belonged to categories for whom a small number of training documents were available.

In another work by Scott and Matwin [93] a new representation of text was proposed based on WordNet synonyms, hypernyms, and POS information [117]. This was used to change the representation of the text from BOW to hypernym density. The new hypernym density representation is computed for each word in the datasets by assigning a POS tag where all nouns and verbs are looked up in WordNet. A list of all synonym and hypernym concepts is assembled, and the infrequent concepts are discarded. The density of each concept is computed by dividing the number of occurrences of the concept in the WordNet output by the number of words in the document. In the study, a Ripper classifier, which is a rule-based ML algorithm, is used to classify six sets of documents with two categories that are created from three different datasets: Reuters-21578 [118], USENET [119], and the Digital Tradition [120]. A comparison between the BOW model and the new hypernym density representation shows that it works well for texts that use an extended or unusual vocabulary, or are written by multiple authors employing different terminologies.

In 2001, a study by Benkhalifa [121] used WordNet concepts as extra features to represent each document in Reuters-21578. The study extended the original BOW model for the document using a strategy called Adding Concepts (AC) which involves adding the concepts identified in the document to its BOW model as extra features. Semi-Supervised Fuzzy c-Means (ssFCM) was used to classify documents from the 115 categories of the Reuters collection. It is experimentally validated that the integration of WordNet concepts helps ssFCM to improve its performance.

Using WordNet concepts alone to represent text does not result in any significant categorization improvement as confirmed in [122]. The study compares the categorization accuracy of classifiers based on the BOW and BOC. The document collection used in the study is a subset of the annotated Brown Corpus semantic concordance [123]. This conclusion agrees with others such as Hotho et al. [99]

that used WordNet concepts alone to represent text. In that Hotho study, three strategies were compared for using concepts as features for text representation such as using only concepts to represent documents such as BOC, AC as complementary features to the BOW model, and replacing terms (words) with concepts (RTC) in the BOW model. The study shows that AC yields better accuracy performance results than the other two methods. The results of this study show that representing documents with only concepts is insufficient as WordNet does not cover all special domain vocabularies. Furthermore, WordNet is limited as it is a manually constructed dictionary and therefore laborious to maintain.

To deal with this problem, other researchers have tried replacing WordNet with other KBs derived from the internet, such as ODP and Wikipedia. For example, in [124] the ODP is used as a knowledge source to provide concepts for text representation. The study used a feature generator technique which searches for new features that describe the target document better than the ones supplied with the training documents. The feature generator constructs new features from ODP categories and adds them to the BOW model using AC strategy. Each ODP node associated with a collection of texts which describes the ODP category (concept). The feature generator uses these ODP categories to learn the definition and scope of the concept. To represent documents using ODP concepts, for each text fragment (such as a word, sentence, paragraph, or the whole document) a feature generation is performed at each of these levels of text fragments. By doing so, various aspects in the document are covered at several levels by a set of concepts which overcome the misleading that may occur when the entire document is considered. The SVM learning algorithm is used to categorize documents whose representations models are enriched by ODP concepts. Experimental results confirm an improved performance of the classification system in comparison with the BOW model.

ODP has a number of drawbacks; the topics are not equally covered, some topics are repeated in different branches of the hierarchy concepts, and sometimes some topics are more influenced by the views of the editors in charge. In order to address these shortcomings, Gabrilovich and Markovitch [85] used Wikipedia as a KB to provide concepts to extend the BOW representation of documents using the feature generator technique proposed in previous work [124]. Using Wikipedia concepts to represent documents using the AC strategy improves the performance

of classification for the 20 Newsgroups by 1.0% more than the BOW model whereas using ODP concepts there is only a 0.5 improvement. The study indicates that using Wikipedia concepts is particularly useful for classifying short documents where several datasets of short documents from the test collections are derived. The results of the experiment yielded greater improvement on short documents than on average-sized and large documents by 7%.

Wikipedia as a knowledge source has many features that make it a good source for concepts. In addition, Wikipedia includes semantic relations such as hyponym, synonyms, and associated concepts between concepts which are captured by hyperlinks between Wikipedia articles (concepts). These semantic relations are also used to enrich document representation in the ATC system, e.g. see Wang [125]. The study indicates that adding synonyms to the BOW model is not useful for ATC, whereas adding the top 5, 10, 15, 20, and 25 associated concepts is useful for classifying documents. The best performance was achieved by adding the best 5 associated concepts to the BOW model as it yielded a 3.6% improvement over BOW model. In the study, hyponyms, which is the category name for each candidate concepts, are also added as extra features to extend the BOW model. The authors compare enriching the document representation with homonyms of candidate concepts from different levels such as first, second, and up to 5 levels. The results of these experiments showed that adding direct hyponymies and hyponymies within the first two levels achieves the best accuracy in the ATC system, and the best result is achieved by adding direct hyponymies which improves the classification accuracy by 4.7%. The study also extends the BOW model for document representation with direct hyponymies and the top 5 associated concepts for each candidate concept. This strategy achieves the best improvement on categorization compared to others by a 4.9% improvement in comparison with the BOW model.

3.6 Combined words and concepts strategies

Using concepts as features for text representation can overcome some of the limitations of other features such as words, phrases, and N-grams. Concepts have many advantages over words and seem to be suitable replacements, especially for synonym and polysemy problems. However, the efficiency of using concepts in

comparison to words as representation features for text in ATC might be reduced as some KBs did not suit the input texts. Therefore, many studies conclude that it is more beneficial to combine words and concepts to represent text and different strategies are implemented such as:

- **Adding Concepts (AC):** This strategy involves adding the concepts identified in the document to its BOW model as extra features and was first used by Benkhalifa [121]. This strategy has since been used by a number of other researchers who demonstrated its improving impact on the performance of English TC [85, 90, 93, 99, 124-126]. Figure 3.1 shows a sample text that discusses the subject of document classification which we have used to illustrate how the AC combined representation model works.

Document classification or document categorization is a problem in natural language processing, library science and information science. The task is to assign a document to one or more classes or categories. This may be done ‘manually’ (or ‘intellectually’) or algorithmically. The documents to be classified may be texts, images, music, etc. A classifier ask himself: ‘Under which descriptors should this entity be found?’ and ‘think of all the possible queries and decide for which ones the entity at hand is relevant’

Figure 3.1 Sample text

By using the BOW representation model, a number of features are extracted such as “document”, “categorization”, “natural”, “language”, “query”, “algorithm” as can be seen in Table 3.1.

Table 3.1 BOW text representation for the text in Figure 3.1

BOW	
f_1	document
f_2	categorization
f_3	natural
f_4	language
f_5	query
f_6	algorithm

where, by using BOC representation model, a number of concepts are identified in the text such as “search engines”, “document classification”, “natural language processing” and “information retrieval”; as can be seen in Table 3.2.

Table 3.2 BOC text representation for the text in Figure 3.1

BOC	
f_1	search engines
f_2	document classification
f_3	natural language processing
f_4	information retrieval

By using the AC strategy to enrich text representation, the BOW model in Table 3.1 is extended by the concepts from Table 3.2 as can be seen in Table 3.3.

Table 3.3 AC text representation for the text in Figure 3.1

AC	
f_1	document
f_2	categorization
f_3	natural
f_4	language
f_5	query
f_6	algorithm
f_7	search engines
f_8	document classification
f_9	natural language processing
f_{10}	information retrieval

- Replacing Terms with Concepts (RTC). This strategy is similar to AC, but words in the document for which a concept has been identified are removed from the combined text representation model. The strategy was first proposed by [99] for English text clustering.

For example, by using the RTC strategy to represent the same sample text that previously represented by BOW and BOC models in Table 3.1 and 3.2; all concepts from the BOC are added it to the RTC representation model and only words which do not have corresponding concepts are added to the RTC representation model such as “algorithm” as shown in Table 3.4.

Table 3.4 RTC text representation for the text in Figure 3.1

RTC	
f_1	algorithm
f_2	search engines
f_3	document classification
f_4	natural language processing
f_5	information retrieval

Unfortunately, the RTC strategy did not use as much as AC strategy which is a result of the lower accuracy that it achieved in [99] with a comparison of AC combined representation model.

- **Adding Concepts and Categories (ACC):** In this strategy, the parent categories of concepts are added to the combined feature vector of the representation model, along with concepts and words. This strategy is similar to what Wang developed in [125, 126], the difference is that only first level parents are used. For example, the concept “document classification“ has the following parent categories “Machine learning”, “Knowledge representation”, “Natural language processing”, “Information science”, “Data mining”. Table 3.5 shows the resulting text representation for the sample text in Figure 3.1 using the ACC combined model.

Table 3.5 ACC text representation for the text in Figure 3.1.

ACC	
f_1	document
f_2	categorization
f_3	natural
f_4	language
f_5	query
f_6	algorithm
f_7	search engines
f_8	document classification
f_9	natural language processing
f_{10}	information retrieval
f_{11}	Computational Linguistics
f_{12}	Machine learning,
f_{13}	Knowledge representation,
f_{14}	Information science
f_{15}	Data mining

3.7 Summary

In this chapter, we discussed how text representation is an important aspect of a successful ATC system and how using different types of features in text representation have a significant impact on the performance of ATC systems. Different types of features used in text representation in ATC systems were reviewed and discussed namely single words, N-grams, and phrases. Furthermore, concepts which provide a unique meaning and behave as a unit of knowledge used as features for text representation model are considered. The different characteristics of KBs were reviewed, and the method by which a concept is determined in free text is analysed. Practically, both WordNet and Wikipedia as KBs were reviewed and it was determined, through a discussion of their

characteristics, that they are ideal resources to provide concepts for text representation.

What makes these KBs ideal for this study is that they are generic, domain independent, have good coverage, and provide rich information about how concepts are related and how strong the relationships are, which makes them ideal resources to provide concepts for text representation.

Although concepts have many advantages over words and seem to be a suitable replacement, studies have shown that it is better to combine words and concepts in text representation. Different strategies that have been applied to use both words and concepts to represent text in ATC systems are reviewed and discussed.

CHAPTER 4

Enhancing the Combined BOW & BOC Representation

4.1 Introduction

Concepts are less ambiguous than N-grams and more specific than words and overcome the low performance of phrases when used to represent text. A concept is a unit of knowledge that provides a unique meaning. It has many advantages over words and seems to be a suitable replacement for the problem of synonymy and polysemy in words. For example, a concept involves mapping synonyms which share the particular unique meaning that the concept provides. In addition, the multiple meanings of polysemous words are mapped to different concepts corresponding to their intended meanings based on the surrounding text. However, using concepts alone does not provide full coverage for the text. Therefore, many strategies are implemented to combine words and concepts to enrich text representation in an ATC system. In this chapter, new developed strategies which enrich the text representation in an ATC system are introduced and compared to other strategies that were previously used.

4.2 Combined Strategies

As discussed in Section 2.2, the classification process begins by passing all the documents in the dataset through a text pre-processing component where texts are cleaned and processed. The result of this component is a set of well-defined features. These features are then used in the text representation component as shown in Figure 4.1. In our ATC system, we have two types of features, words and concepts, concepts in our system are provided by Wikipedia where each article title is the descriptor of the concept. By using each type alone, we get the BOW and BOC model respectively. Both the BOW and BOC models represent the features in the VSM based on the appearance of the feature in the document. As it appears in the document, the corresponding value in the VSM is the weight of the feature in

the document using the TFIDF weighting scheme (see Section 2.2.2); otherwise it takes the value zero. Both types of features are also used in combined models which are built using different strategies such as AC, RTC and ACC which were proposed and used by others [85, 99, 126] (see Section 3.6.); and new developed strategies, Adding Unmapped Concepts (AUC), and Using Concepts for Terms which do not appear in the Document (CTD) which will be described with more details in this chapter. As can be seen in Figure 4.2, each strategy organises features in the VSM model in different ways to represent text in the ATC system, which leads to different representation models for the documents.

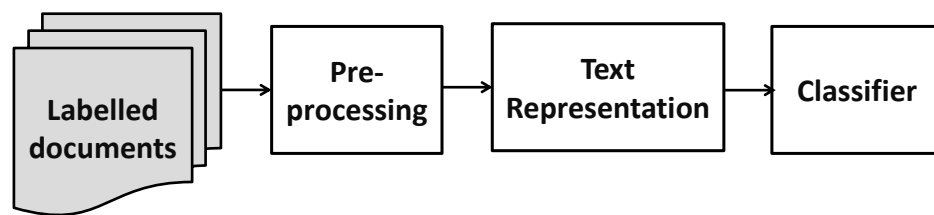


Figure 4.1 The three main components of an ATC system

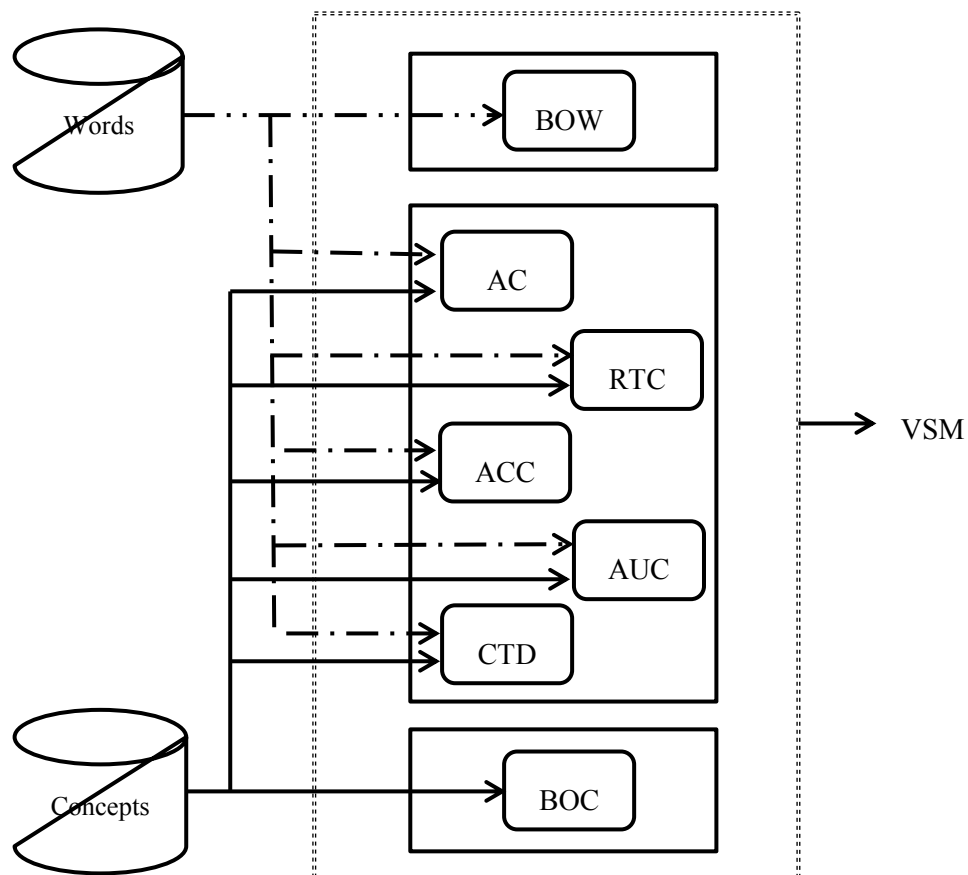


Figure 4.2 Text representation component in our ATC system

The following example illustrates the use of the above mentioned strategies for representing a sample document. A document from the Classic 3 dataset, “cran.001051”, (see Section 5.2.1) that belongs to the category CRAN discusses the stability of thin-walled non-stiffened circular cylinders under axial compression in the design of high-speed aircraft including the effects of internal pressure, as can be seen in Figure 4.3.

Each document in a dataset has general and specific words. The general words are often distributed over various documents which belong to different categories, whereas specific words are only shared by documents that belong to the same category. In some cases, these specific words can appear with more density than other words in a document. For example, in the document in Figure 4.4, the words “axial”, “compression”, and “cylinder” are considered to be significant words with a term frequency of 3, 6, and 7 respectively.

Document	<i>The stability of thin-walled unstiffened circular cylinders under axial compression including the effects of internal pressure. The stability of thin-walled unstiffened circular cylinders under axial compression including the effects of internal pressure. In the design of high-speed aircraft the importance of unpressurized and pressurized monocoque cylinders necessitates a reliable analysis procedure for the compressive buckling of cylindrical shells. Analysis by the classical small-deflection theory has proved inadequate. Recent large-deflection theoretical treatments of the problem have shown reasonable correlation with experiments but require a prior knowledge of the initial imperfections of the cylinder. Developed in this paper is a semi empirical procedure which permits a compressive buckling analysis of cylindrical shells with knowledge of the cylinder geometry only. This analysis is achieved by correlating experimental data statistically with theoretical parameters. In order to provide data not previously available, an extensive series of axial compression tests of pressurized cylinders has been performed. These data, together with all other known test data, are analysed semi empirically. In the analysis best-fit curves are presented using theoretical parameters and shapes of curves where applicable. Unpressurized and pressurized cylinder compressive buckling curves are then developed as 90 per cent probability curves from the test data. In general, these statistically defined design curves are significantly lower than previously available design curves</i>
cran.001051	

Figure 4.3 Example of a document from the Classic 3 dataset

On the other hand, some specific words in the document do not appear as frequently as others such as “aircraft” which is only mentioned once in the document in Figure 4.3. Different representation models used to represent this document can be seen in Figure 4.4.

Strategy (No. of features)	Most frequent 20 features
BOW (81)	<i>curv (6.0) pressur (5.0) analysi (5.0) compress (6.0) data (5.0) cylind (7.0) test (3.0) knowledg (2.0) axial (3.0) theoret (3.0) paramet (2.0) effect (2.0) correl (2.0) statist (2.0) develop (2.0) under (2.0) avail (2.0) design (3.0) cylindr (2.0) procedur (2.0) aircraft (1.0) cylind (7.0)</i>
BOC (37)	<i>cylinder (engine) (7.0) theory (4.0) data (5.0) data analysis (5.0) axial compressor (3.0) knowledge (2.0) pressurization (5.0) stress (mechanics) (3.0) design (3.0) compression (physics) (3.0) experimental aircraft (1.0) pressure (2.0) experimental data (1.0) exoskeleton (2.0) buckling (3.0) algorithm (2.0) test data (2.0) trigonometric functions (2.0) deflection (engineering) (2.0) cylinder (geometry) (2.0) cabin pressurization (5.0)</i>
AC (114)	<i>cylinder (engine) (7.0) cylind (7.0) compression (physics) (3.0) theory (4.0) buckl (3.0) compress (6.0) stress (mechanics) (3.0) cabin pressurization (5.0) pressur (5.0) axial compressor (3.0) design (6.0) curv (6.0) axial (3.0) high-speed flight (1.0) aircraft (2.0) algorithm (2.0) data analysis (5.0) unstiffen (2.0) knowledg (2.0) exoskeleton (2.0)</i>
RTC (35)	<i>experimental data (1.0) cylinder (geometry) (2.0) stress (mechanics) (3.0) statistics (2.0) previous (2.0) flight dynamics (2.0) cross-correlation (1.0) avail (2.0) design (3.0) cabin pressurization (5.0) test data (2.0) knowledge (2.0) theory (4.0) data analysis (5.0) axial compressor (3.0) trigonometric functions (2.0) semi (2.0) pressure (2.0) buckling (3.0) unstiffen (2.0)</i>
ACC (201)	<i>aircraft (3.0) compress (6.0) axial (3.0) knowledge (3.0) algorithm (2.0) elasticity (physics) (2.0) high-speed rail (2.0) data (13.0) fluid dynamics (2.0) classical mechanics (3.0) analysi (5.0) aerospace engineering (2.0) pressure (3.0) engineering thermodynamics (2.0) curv (6.0) data analysis (6.0) cylind (7.0) geometry (2.0) covariance and correlation (2.0) theoretical computer science (1.0) compression (physics) (3.0) elementary shapes (1.0)</i>

Figure 4.4 Different representation models for the document “cran.001051”

Each representation model shows the significant features in the document and their number of frequency. By using both BOW and BOC models as can be seen in Figure 4.4, words with less density such as “aircraft” is mentioned once in both models and does not reflect the significant meaning that it carries like other significant words. Whereas in combined models using the strategies AC and ACC the term frequency of the word “aircraft” is increased in the document by 2 and 3 respectively.

In general, the number of concepts identified in a text is always less than the number of words which leads to a smaller representation size. For example, for a given example the document “cran.001051” in Figure 4.3, the BOW representation model has 81 words as can be seen in Figure 4.4 where the BOC representation model is smaller than the BOW model which only has 37 concepts. But when both features are combined using these strategies, the resulting representation model is larger than the BOW and BOC models such as in AC and ACC with 114 and 201 features respectively. In this thesis, the largest representation model is built using the ACC strategy which only contains categories that are first level parents of concepts that result in less features than what was proposed by [125, 126]. The smallest representation model is built using the RTC strategy with only 35 features, which is less than BOW and BOC, but it suffers from inefficient accuracy when it is used in the ATC system. Therefore, two strategies are developed to reduce the size of the representation model without affecting the accuracy of the classification task negatively.

4.2.1 Adding Unmapped Concepts (AUC)

This strategy simply involves taking the basic BOW and adding to it the concepts whose representative words do not explicitly appear in the BOW model. For example, “infant” is a concept mentions in the Classic 3 dataset which is mapped to a number of words and terms such as “baby”, “new born”, “early childhood”, “young child”, “little child”, and “little one”, including the word “infant”. In some documents, the term “young child” presents where in other documents you may find synonyms such as “new born” or a word “baby”. By adding the concept “infant” to these documents, the BOW model is enriched with the same concepts

and these related documents which do not share similar words become more closely related.

This strategy first involves creating a “Concept-Words Map” for the whole dataset. To achieve that, for each concept identified in the dataset, the corresponding word(s) that share the unique meaning provided by the concept are mapped to it. By doing this, the algorithm will resolve synonyms and capture different words which refer to the same concept in the documents of the training subset. These words are considered as alternative labels for the concept in the “Concept-Words Map”. The concept label provided by the Knowledge Base (KB) is considered as a preferred label. In order to build a combined model to represent a given document using the AUC strategy, the following tasks have to be performed. Firstly, a BOW vector has to be created for the document, and all words that are considered as features in the BOW model are added to AUC model’s vector. In this way, words which have a significant frequency value in the BOW model and carry an important value for the classification task are kept in the AUC model. Then, for each concept in the BOC model of the document, the “Concept-Words Map” is checked for alternative labels of the concept, in case one of them is appearing in the document. If one of the alternative labels appears in the text, the preferred label for the concept is added to the AUC model. Otherwise, nothing is added regarding that concept, and we move to the next one, as can be seen in Figure 4.5. The difference between the AUC strategy and AC strategy is that only concepts mapped to different alternative labels are added to the combined representation model.

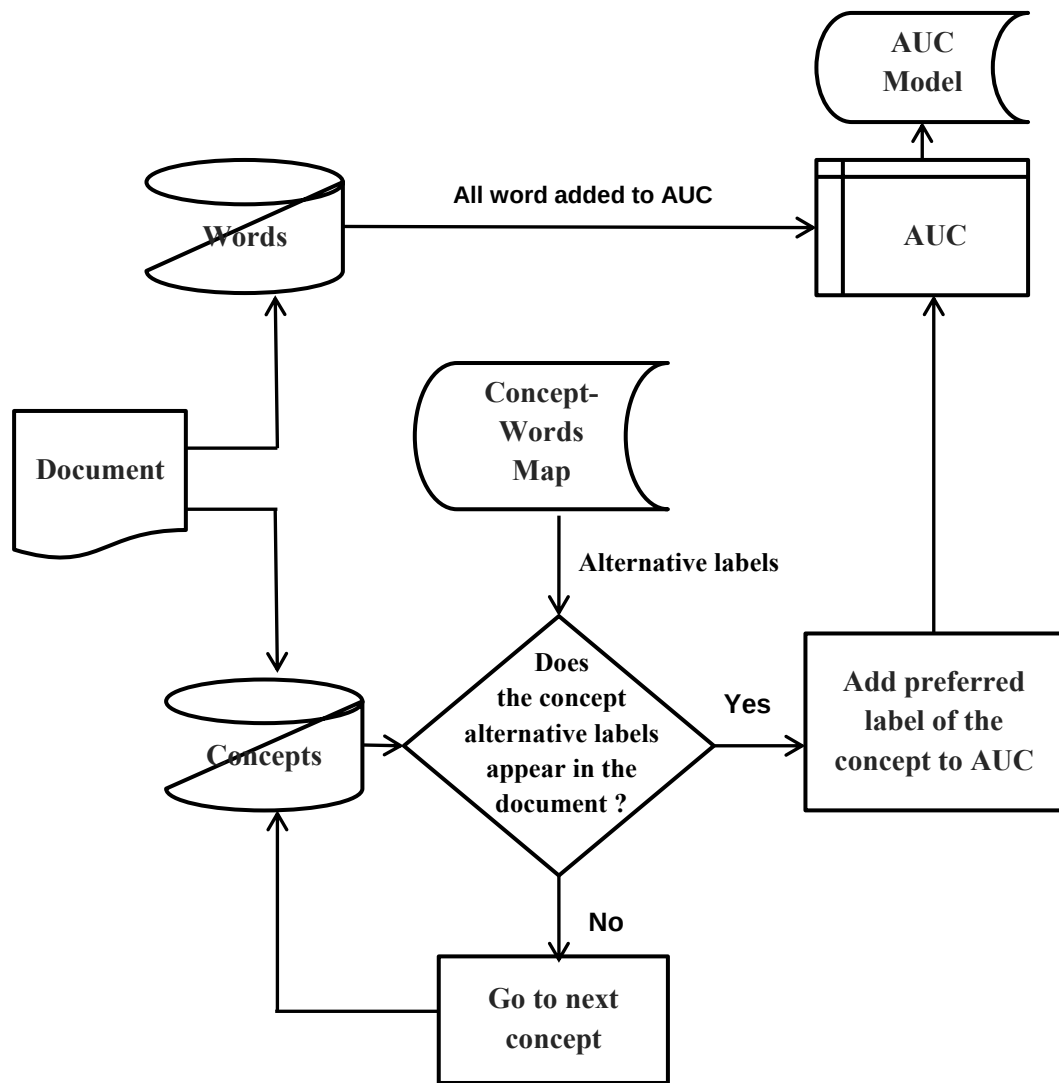


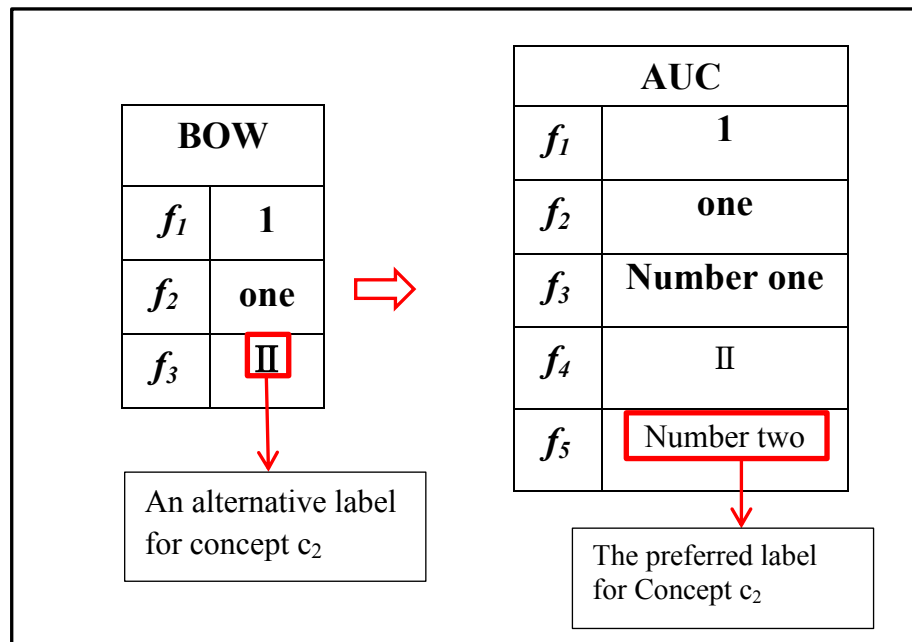
Figure 4.5 Process of building AUC model

Generally, to represent documents using AUC strategy, first the Concept-Words map needs to be created for the dataset. For example, consider a sample dataset D , which contains a set of documents, and its corresponding Concept-Words map as shown in Table 4.1. Representing a document, $d_i \in D$, using BOW model results in the feature set $\{1, \text{one}, \text{II}\}$, as illustrated in Figure 4.6. By using the AUC strategy to build the combined representation model for d_i , first all features from BOW model are added to the AUC model as can be seen in Figure 4.6.

Table 4.1 Concept-Words Map for the given example

Concept-Words Map			
Concept	c_1	c_2	c_3
Preferred label	Number one	Number two	Number three
Alternative Labels	1	2	3
	one	two	Three
	I	II	III

Then for each concept identified in the document, we need to check the Concept-Words map (see Table 4.1) of the dataset to find the “alternative labels” that appear in the document. From the map, we can see that both concepts c_1 and c_2 alternative labels (i.e. II is an alternative label for concept c_2) are appearing in the document d_i . As a result, the preferred label of both concepts c_1 and c_2 (“Number one” and “Number two”) are added to the AUC combined model representation as can be seen in Figure 4.6.

**Figure 4.6 The AUC representation model for the given example**

For the document “cran.001051”, the AUC strategy builds a representation model that increases the size of the representation model for BOW and BOC models by enriching the BOW model with concepts that bring related documents closer, as

can be seen in Figure 4.7. The word “aircraft” which is mentioned once in the BOW model and once as a concept, “experimental aircraft”, in the BOC model, have less density which does not reflect the significant meaning that it carries like other significant words with high density in the representation, as can be seen in Figure 4.4. The AUC strategy makes the representation model benefit from both concept and word and improves the density for this significant feature.

In addition, the AUC strategy increases the number of features in comparison to the combined model built using the RTC strategy and decreases the size of the combined model in comparison to the one built using the AC and ACC strategies as can be seen in Figures 4.4 and 4.7.

Strategy (No. of features)	Most frequent 20 features
AUC (98)	<i>cylind (7.0) aircraft (1.0) cylinder (engine) (7.0) compress (6.0) buckl (3.0) curv (6.0) pressur (5.0) analysi (5.0) compression (physics) (3.0) axial (3.0) algorithm (2.0) data (5.0) correl (2.0) deflection (engineering) (2.0) paramet (2.0) design (3.0) knowledg (2.0) experimental aircraft (1.0) parameter (2.0) general officer (1.0) classical mechanics (1.0)</i>

Figure 4.7 The combined representation using the AUC strategy for the document “cran.001051”

The AUC strategy keeps words in the representation model, and only concepts that did not appear as a part of the BOW model are added. When documents between categories share a lot of features, the classification task becomes harder, and a lot of documents are misclassified. By using the AUC strategy, the significant features in documents which are less frequent in comparison to others are increased, and concepts that are mentioned as synonyms between documents are kept. In this case, when using the AUC strategy to build the combined representation model in the ATC system, it reduced the number of misclassified documents.

For example, the WebKB dataset has 4199 documents that belong to four categories (see Section 5.2.3); these documents share a lot of features between the four categories. Using the SVM ML algorithm as a classifier, these documents have been classified using our ATC system. The results of the classification are

presented in Figures 4.8 - 4.13 using scatter plots. Table 4.2 gives legend descriptions of the plot symbols used in Figures 4.8- 4.13.

Table 4.2 Meaning of legend in scatter plot

symbol	meaning
x	Document is corrected classified in Class1
□	Document is misclassified in Class1
x	Document is corrected classified in Class2
□	Document is misclassified in Class2
x	Document is corrected classified in Class3
□	Document is misclassified in Class3
x	Document is corrected classified in Class4
□	Document is misclassified in Class4

According to the Table, each category is represented with a different colour such as class 1 with blue, class 2 with red, class 3 with green and class 4 with turquoise as can be seen in both vectors in the scatter plot where one represent the actual class and other the predicted class. Each Figure shows a representation of misclassified and correct classified documents when each representation model is used in ATC system. The correct classified documents are represented by an “x” when both actual class and predict class are agreed. On the other hand, the misclassified documents are represented as squares when actual class and predict class are disagreed.

From Figures 4.8- 4.13 it can be seen that the number of incorrectly classified documents is 602, 953, 619, 850, and 594 for BOW, BOC, AC, RTC, and ACC respectively, where the representation model that used the AUC strategy misclassified only 579 documents. The lower numbers of misclassified documents are achieved by AUC and ACC combined representation models in comparison to others. In addition, from Figure 4.13 it can be seen that the number of documents that misclassified from class 1 to class 4 are less with AUC combined representation model than ACC combined representation model in Figure 4.12

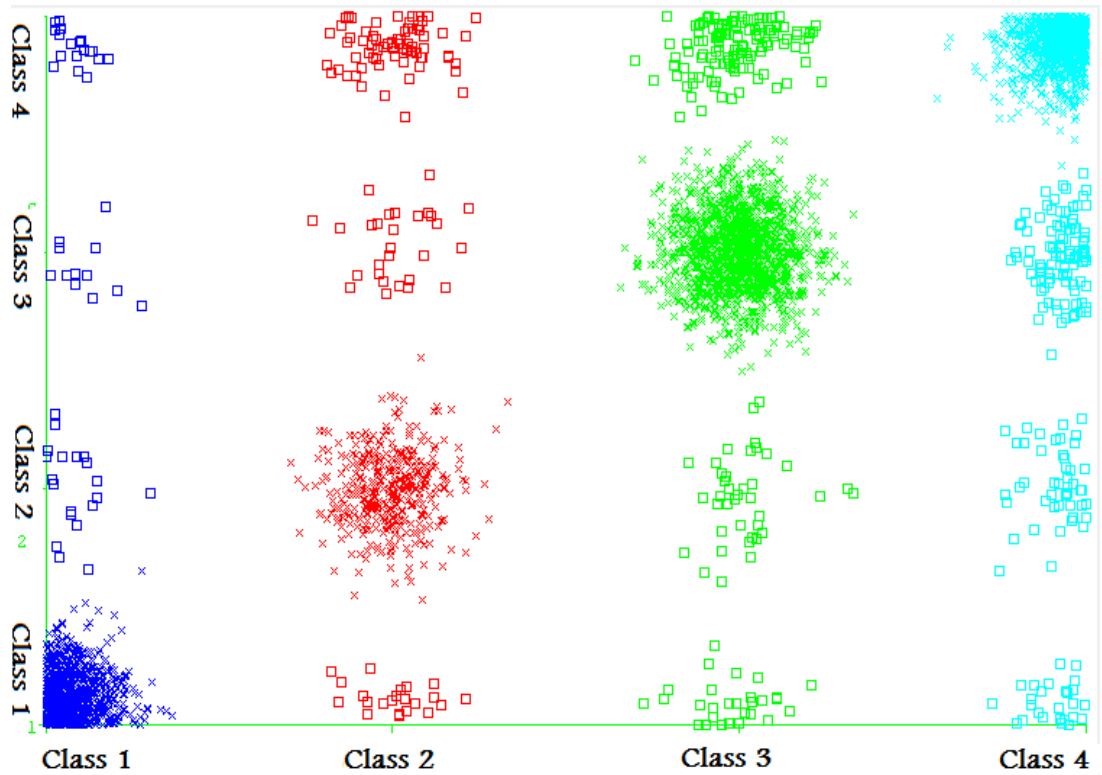


Figure 4.8 The 602 misclassified documents of the WebKB dataset with BOW representation model in ATC system

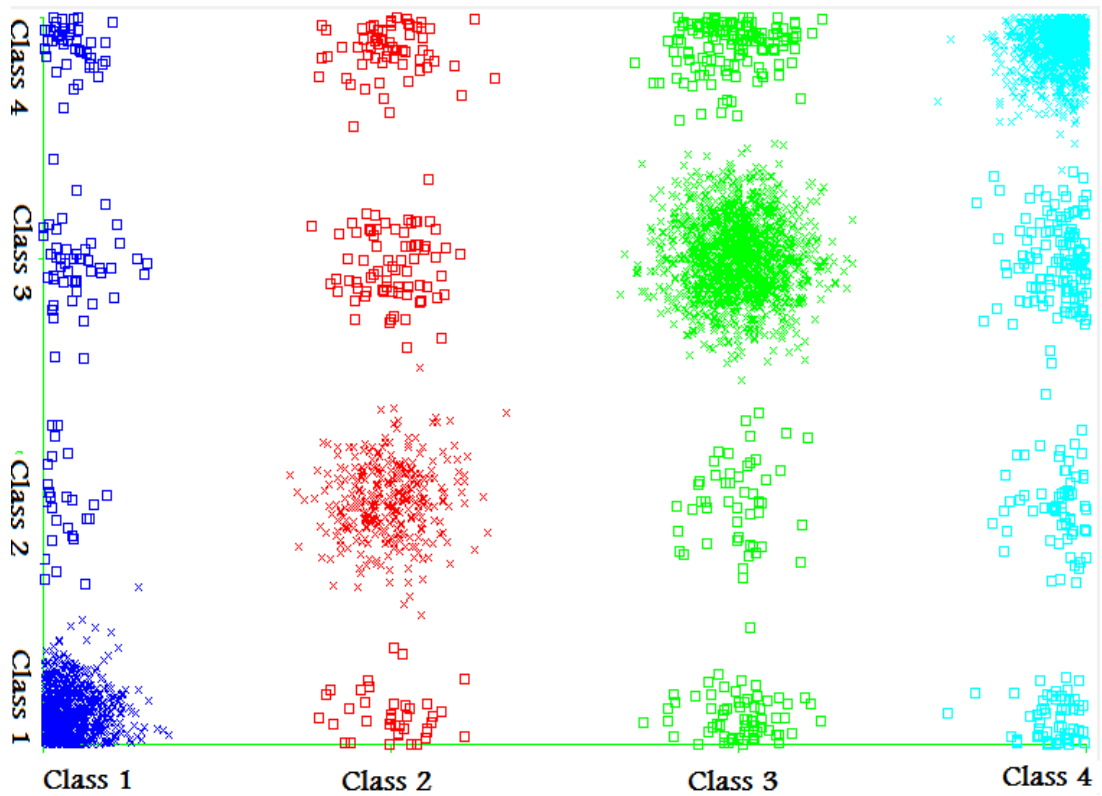


Figure 4.9 The 953 misclassified documents of the WebKB dataset with BOC representation model in ATC system

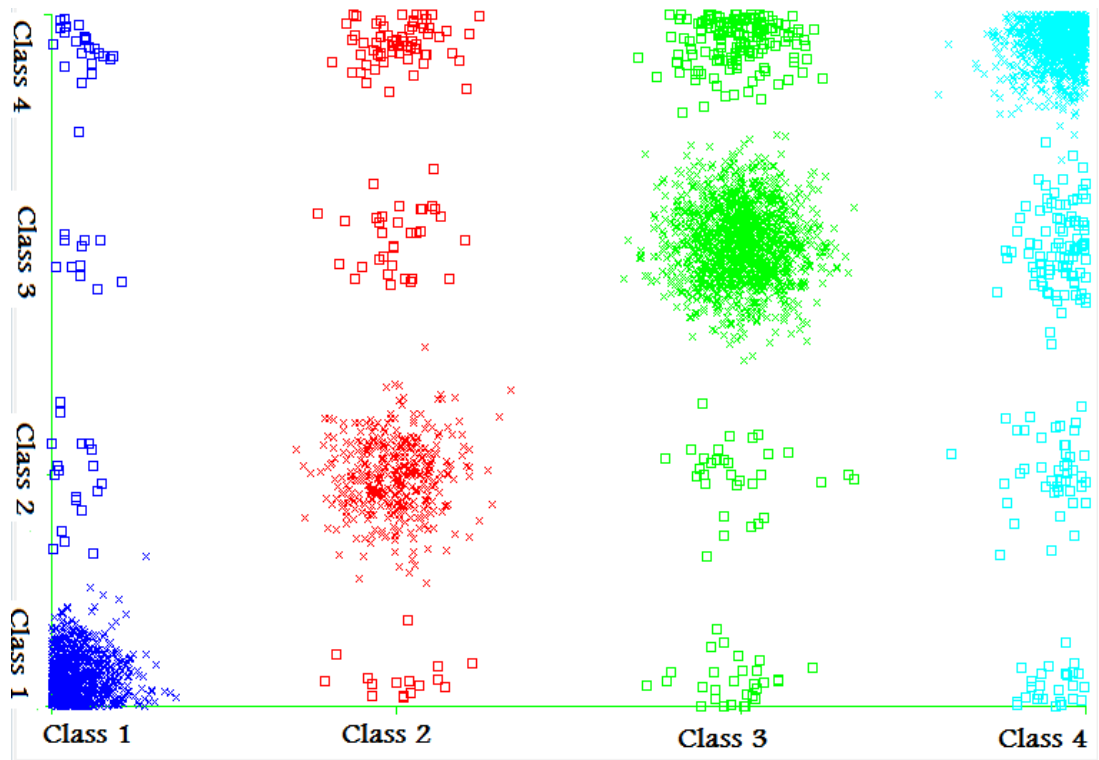


Figure 4.10 The 619 misclassified documents of the WebKB dataset with AC representation model in ATC system

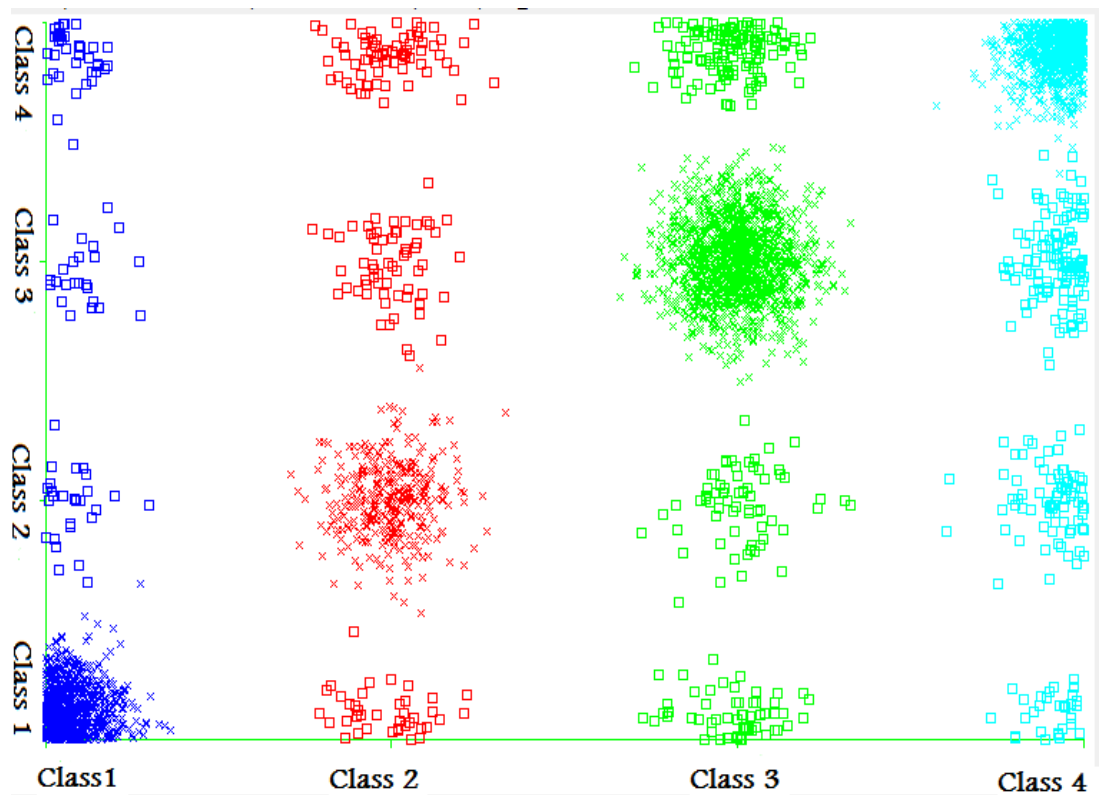


Figure 4.11 The 850 misclassified documents of the WebKB dataset with RTC representation model in ATC system

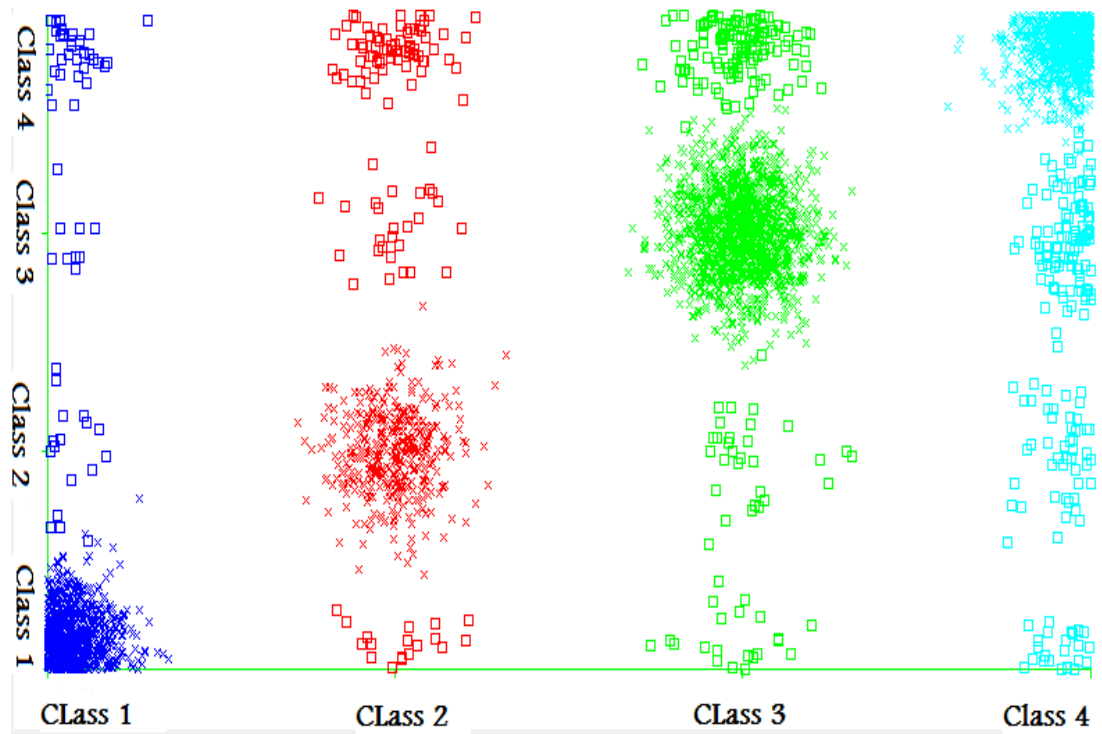


Figure 4.12 The 594 misclassified documents of the WebKB dataset with ACC representation model in ATC system

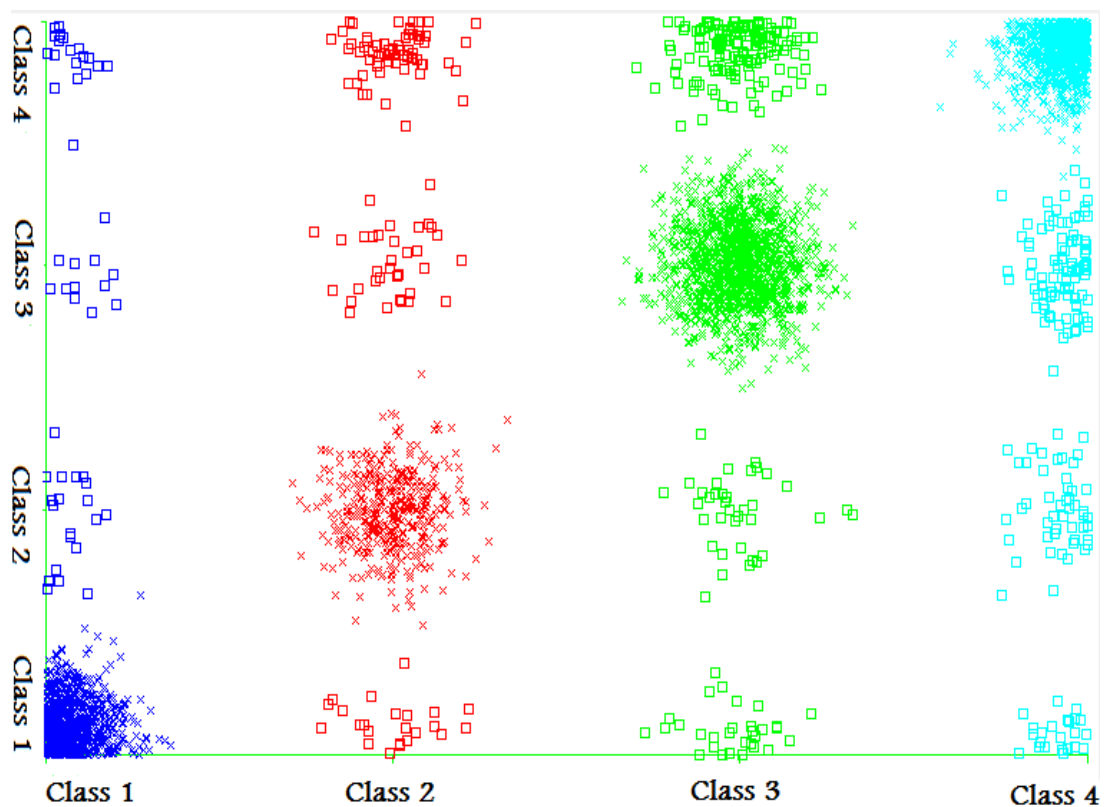


Figure 4.13 The 579 misclassified documents of the WebKB dataset with AUC representation model in ATC system

4.2.2 Using Concepts for Terms which do not appear in the Document (CTD)

All previous strategies used in the text representation component to build a combined model used both concepts and words with their corresponding weight in the combined VSM. As a result, the numbers of features in the combined models are larger than the number of features in the BOW and BOC models individually. The CTD strategy does not change the original dimensions of the BOW model. Instead, this strategy is similar to the BOW model in two ways: the size of the VSM and the type of features that are used to represent a document, where each dimension in the VSM corresponds to a word from the BOW dictionary.

The strategy works like the BOW model for words that appear in the document, where their corresponding dimensions in the VSM will have the value of the words' TFIDF weight. The only difference in this strategy is for those words that do not appear in the document but have a corresponding concept in the document's BOC model. For these words, the weight of their related concepts in the document will be used as corresponding dimensions in the VSM. The CTD strategy works as follows. First, a "Word-Concepts Map" needs to be created for the whole dataset. To achieve that, for each word that has been identified by a concept(s), the word is mapped to its corresponding concept(s). By doing this, the algorithm will resolve synonyms and capture different words which refer to the same concept in all the documents of the training subset. In addition, words that have multiple meaning are mapped to different concepts. To represent a document using the CTD strategy and build the combined model, first all words which are considered as features in the BOW model dictionary are checked for their appearance in the document. In the case that a given word appears in the document, the word's TFIDF weight in the document is used in the CTD model, as can be seen in Figure 4.14. If the word does not appear in the document, the "Word-Concepts Map" is checked to see if that word has a corresponding concept. In the case that the word has a concept, the concept is retrieved from the "Word-Concepts Map" and then checked if it is identified in the document's BOC model. In the case that the concept exists, the word's corresponding dimension in the CTD model will be assigned to the weight of the concept. If the corresponding

concept does not exist, the word will not be represented in the CTD combined model as can be seen in Figure 4.14.

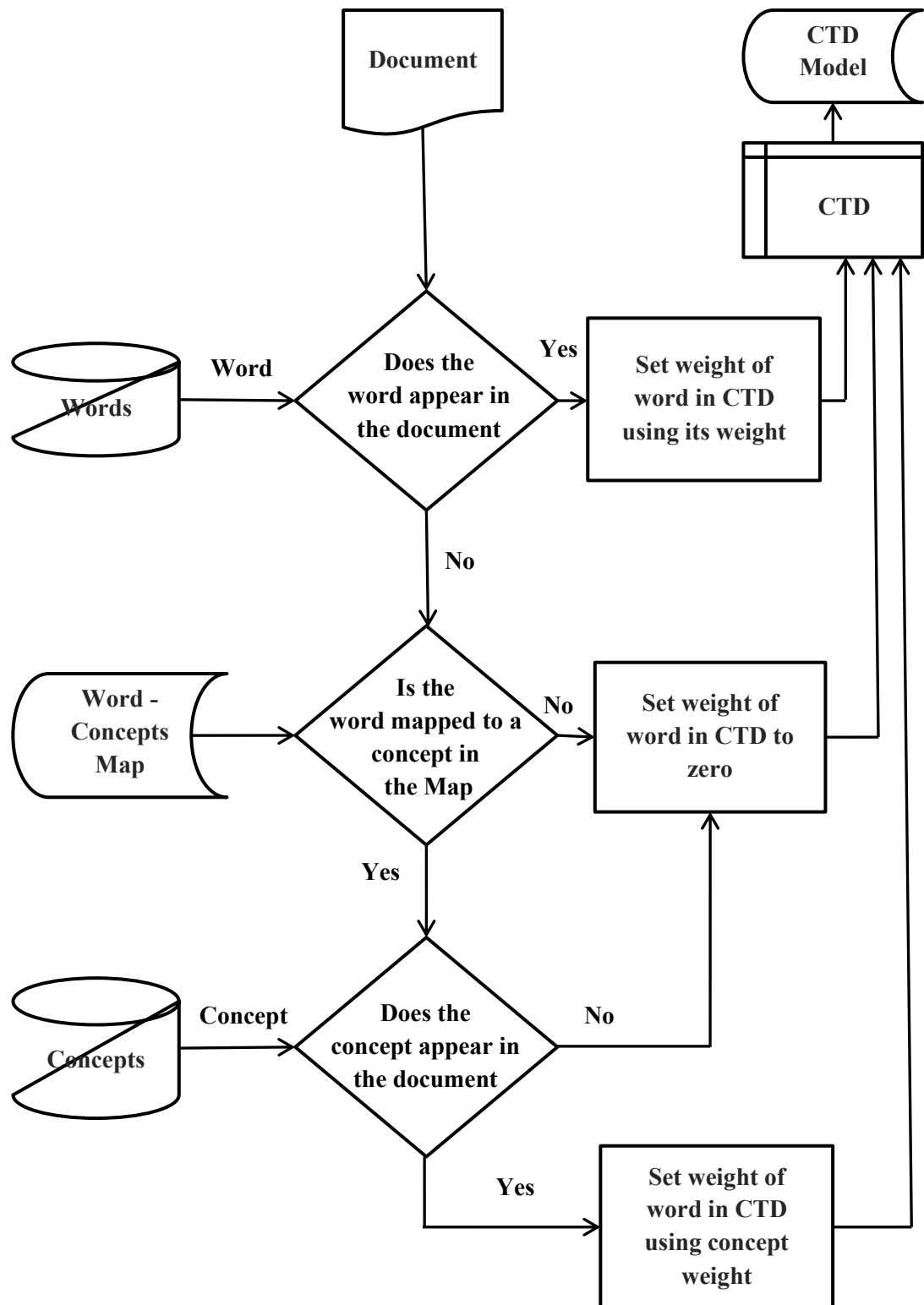


Figure 4.14 process of representing document using CTD strategy

As an example, consider a dataset D , which contains two documents, d_1 and d_2 , where d_1 has the feature set {Two} and d_2 has the feature set {one, II}, as can be seen in Figure 4.15. The BOW model is used to represent these documents as shown in Figure 4.15 where both documents share the same meaning of the concept “Number two” but no features are shared.

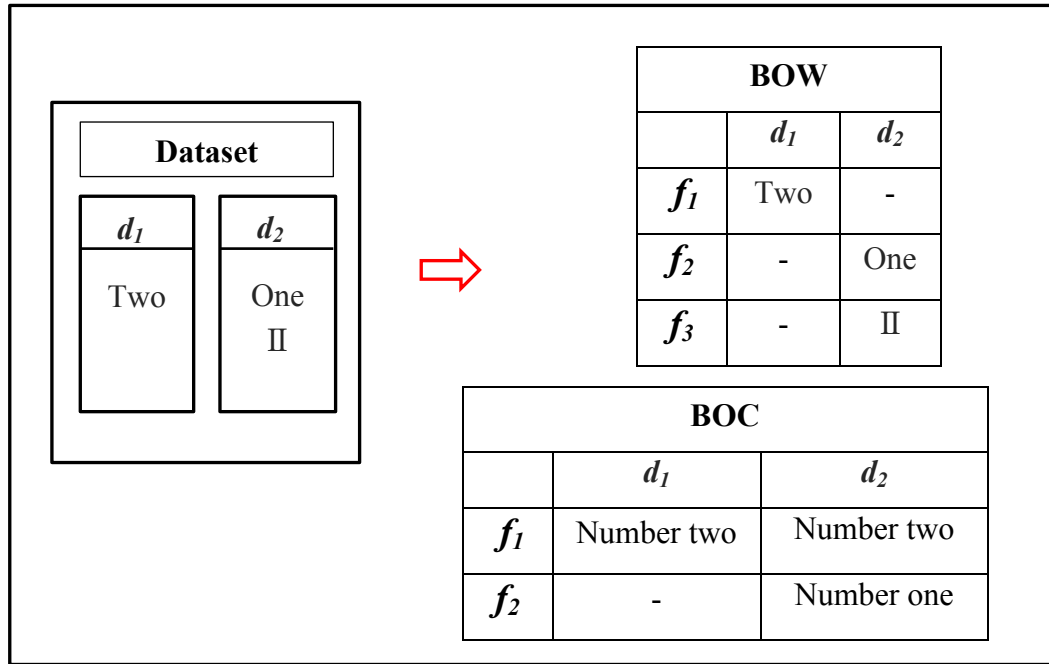


Figure 4.15 The BOW & BOC model for the documents that are belonging the dataset

Using the CTD strategy, this problem is solved by creating a combined representation model for the document as follows. First, the Word-Concepts map needs to be created for the dataset as can be seen in Figure 4.16. In the map, both words “II” and “Two” are mapped to the concept “Number two” as these words share the same unique meaning of the concept.

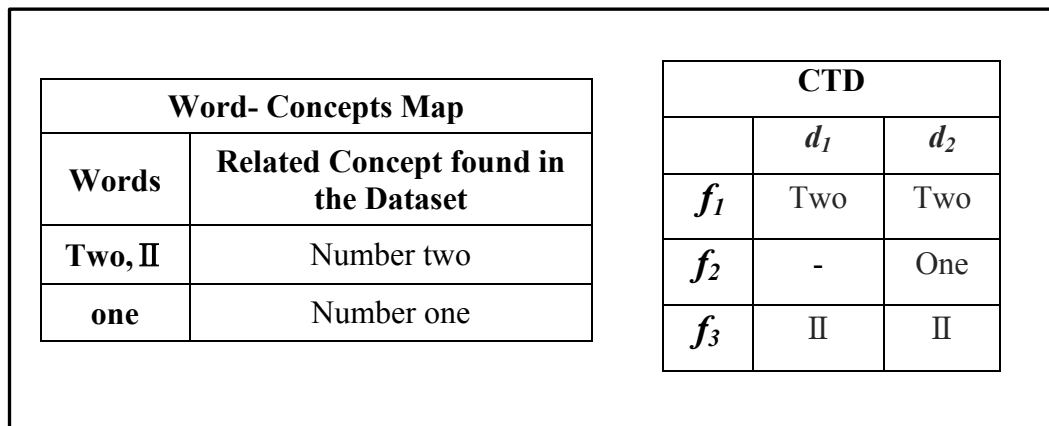


Figure 4.16 CTD representation model for the given example

The document d_I is then represented using CTD strategy where all words that are considered as features in BOW model dictionary are checked for their appearance in the document. In case the word appears as “Two”, it will be represented in the CTD model. Otherwise, the word will be checked in the Word-Concepts map if it has related concept(s) or not. For example, the word “II” does not appear in document d_I , but it is mapped to the concept “Number two”. The concept “Number two”, in turn, appears in the BOC model for document d_I , as can be seen in Figure 4.15. As a result, the word “II” will be represented in CTD model for document d_I as shown in Figure 4.16. Accordingly, in the CTD model, words that share the same meaning are represented in the model, and the size of the CTD model does not get bigger than the size of BOW model.

Using the CTD strategy in the text representation component, more words from the dictionary are represented in the VSM in comparison to the BOW. This means documents which discuss the same topics can share similar words, leading to a relatively higher classification accuracy compared to using the BOW. For example, given the document “cran.001051” (see Figure 4.3), the CTD strategy builds a representation model that increases the number of words identified from the words dictionary in the VSM in comparison to the BOW model. Figure 4.4 shows that the BOW model for the document comprises 81 words, whereas by using the CTD strategy it has 127 words as can be seen in Figure 4.17. The word “fuselag” is not present in the BOW model of the document “cran.001051”, whereas in the CTD model it is represented. This is due to the fact that the CTD strategy has examined the Word-Concepts map and mapped the word “fuselag” to the “cabin pressurization” which also appears as a feature in the BOC model of the document (as can be seen in Figure 4.4).

The Strategy (total number of features)	Most frequent 20 features
CTD (127)	<i>curv (6.0) cylind (7.0) analysi (5.0) analyt (5.0) fuselag (5.0) inform (7.0) rate (2.0) data (5.0) longitudin (5.0) aircraft (1.0) compress (6.0) field (5.0) framework (4.0) pressur (5.0) axial (3.0) compressor (3.0) philosophi (3.0) model (6.0) knowledg (2.0) buckl (3.0) practic (3.0) surfac (2.0)</i>

Figure 4.17 The combined representation using CTD strategy for the document “cran.001051”

By increasing the number of identified words in the representation model, the documents that share similar subjects will share more words in the CTD representation models when compared to the BOW model, and it will increase the number of classified documents in the ATC system. The CTD strategy embeds the concept's weight directly in the weights of the corresponding words which reduces the number of misclassified documents in the ATC using the BOW model.

For nine of the given documents from the Classic 3 dataset, which has three categories (for more details see Section 5.2.1), where documents one, two, and three belong to the category CISI, documents four, five, and six belong to category CRAN and documents seven, eight, and nine belong to the category MED. The CISI contains abstracts from information retrieval papers; CRAN covers abstracts from aeronautical system papers, and MED contains abstracts from medical journals. The full nine documents can be found in the Appendix A, and they are various in length. Three document samples, documents three, six, and eight, can be seen in Figure 4.18.

Document 3 cisi.001260	<i>The Review of Scientific Instruments with Physics News and Views.</i> <i>A study of the periodicals for the subjects of physics and radio has been undertaken with the hope of indicating those which are most used in each field. The method selected is similar to that used by P.L.K. Gross and E.M. Gross.</i>
Document 6 cran.001293	<i>design of stiffened cylinders in axial compression . the problem of optimum design of axially compressed cylinders stiffened by rings and stringers is discussed . particular attention is devoted to configurations suitable for large launch vehicles. consideration is given to the analytical techniques for determining strength as well as the procedures for optimization .</i>
Document 8 med.000801	<i>autopathy: follow-up of cases the publication of two papers recently in the acta paedopsychiatrica by d. arn. van krevelen on the subject of "autistic psychopathy", has stimulated the author to present 5 cases which appear to conform to his description. the only value of this brief contribution lies in the fact that they have all been followed up for at least 10 years, and this may give some idea of the eventual results in this type of cases.</i>

Figure 4.18 Example of three documents from the Classic 3 dataset

From Figure 4.18 it can be seen that these documents are short in comparison to other documents in the datasets and contain fewer words which leads to an easy misclassification of these documents when only words are used in the representation model. The number of words identified in each of the nine sample documents (see Appendix A), using the BOW and CTD representation models, are shown in Table 4.3. For example, in document three only 20 words are used to represent the document in the BOW model wherein the document one more words are identified in the BOW model because of the length of the document. In addition, some words that mention in the document are not used to represent the document as they are rare words and not considered in the BOW dictionary. On the other hand, the number of words identified in the CTD model is increased in all the documents as can be seen in Table 4.3.

Table 4.3 The classification output for the nine documents using both BOW and CTD models

Document No	BOW		CTD	
	prediction	# of words	prediction	# of words
1	F	40	T	41
2	F	47	T	52
3	T	20	T	21
4	F	28	T	32
5	F	40	T	62
6	T	23	T	25
7	F	19	T	22
8	F	20	T	21
9	F	46	T	63

In some cases such as document five and document nine, the CTD strategy identified 20% more words than the BOW models.

When the SVM ML algorithms are used to classify these documents to their correct category using the BOW model only two documents out of nine are correctly classified, document three and document six with a prediction of “True”, as can be seen in Table 4.3. By using the CTD strategy to represent the same nine documents in the ATC system with the SVM ML to classify these documents, the classifier succeeded in predicting the correct category for all the documents, as can be seen in Table 4.3. In addition, the prediction given by the SVM classifier for each document can be seen in Table 4.4. It can be concluded that when the CTD strategy is used to represent documents, it improves the classification task in comparison to the BOW model.

Table 4.4 The SVM prediction for each category with the BOW and CTD models

Document No	BOW			CTD		
	CISI	MED	CRAN	CISI	MED	CRAN
1	0.33	0.00	0.66	0.66	0.0	0.33
2	0.33	0.00	0.66	0.66	0.00	0.33
3	0.66	0.33	0.00	0.66	0.00	0.33
4	0.66	0.33	0.00	0.33	0.0	0.66
5	0.66	0.00	0.33	0.33	0.00	0.66
6	0.33	0.00	0.66	0.33	0.00	0.66
7	0.66	0.33	0.00	0.33	0.66	0.0
8	0.66	0.33	0.00	0.33	0.66	0.00
9	0.66	0.33	0.00	0.33	0.66	0.00

The result of the classification prediction for the SVM ML algorithm when it is used with all the representation models can be seen in Table 4.5. Both the AUC and CTD developed strategies, which combined the words and concepts to enrich the BOW model, succeeded in predicting the correct category for each document of the nine example documents. In addition, document eight which was misclassified by all representation models is correctly classified by both the developed strategies AUC and CTD when used to represent the document in the ATC system.

Table 4.5 Classification prediction of the nine sample documents with different representation models

Document No	BOW	BOC	AC	RTC	ACC	AUC	CTD
1	F	T	T	F	T	T	T
2	F	T	T	T	T	T	T
3	T	T	F	T	T	T	T
4	F	T	T	F	T	T	T
5	F	T	T	T	T	T	T
6	T	F	T	F	F	T	T
7	F	T	F	T	F	T	T
8	F	F	F	F	F	T	T
9	F	T	T	T	T	T	T

4.3 Summary

In this chapter, two new strategies are developed to enrich the text representation in the ATC system. These strategies are based on combining various selected aspects of the BOW and BOC representations in order to enhance the

classification performance further. These strategies are namely Adding Unmapped Concepts (AUC) and the second one uses Concepts for Terms which do not appear in the Document (CTD). Both strategies are introduced and compared to other strategies previously used to enrich text representation and provide an efficient implementation.

CHAPTER 5

English Text Classification using Combined Representation

5.1 Introduction

In this chapter, the proposed strategies to enrich text representation by using combined models that contain both words and concepts are examined with English text in an ATC system. To measure the performance of the system with the proposed strategies, standard English datasets are used and reviewed in detail as is the experiment design. An experimental ATC system has been developed and its performance evaluated and compared to a number of different text representation models; namely the basic BOW model, basic BOC model, and different strategies that combine both words and concepts.

5.2 Experimental Datasets

The performance of the developed strategies is measured in the ATC system by using standard datasets that are widely used in evaluating the text classification algorithms. These datasets include the 20 Newsgroups, World Wide Knowledge Base, and the Classic 3 datasets. Below is a review of these datasets.

5.2.1 Classic 3 Dataset

The Classic 3 dataset collection is a well-known benchmark dataset used in text mining [127]. This dataset consists of four different document collections: CACM, CISI, CRAN, and MED. These documents consist of titles and abstracts of academic papers from four different subjects where CACM contained abstracts from computer system papers, CISI contains abstracts from Information Retrieval (IR) papers; CRAN covers abstracts from aeronautical system papers, and MED contains abstracts from medical journals. All documents are retained, resulting in about 7,098 documents and researchers refer to it as the Classic 4 dataset. In most studies researchers discard the category CACM and refer instead to the Classic 3

dataset; the remaining part of the dataset contains 3,891 documents in three categories.

Table 5.1 specifies the three remaining categories and their sizes. This dataset is used in many text mining applications such as text clustering [128, 129] and text classification [130].

Table 5.1 List of categories and number of documents in the Classic 3 dataset

Classic 3	
Category	Total number of documents
CISI	1460
CRAN	1398
MED	1033

5.2.2 20 Newsgroups dataset

The 20 Newsgroups dataset [131], originally collected by Ken Lang [132], is a collection of approximately 20,000 Newsgroups documents spread across 20 different newsgroups, each corresponding to a different topic.

Each newsgroup contains exactly 1,000 positive examples except ‘soc.religion.christian’, which contains 997 documents. Some of the Newsgroups documents are similar to each other, for example, comp.sys.ibm.pc.hardware and comp.sys.mac.hardware, while others like misc.forsale and soc.religion.christian are unrelated. The 20 Newsgroups is a popular dataset for experiments in text applications of ML techniques such as text clustering [133] and text classification [85, 126]. Table 5.2 specifies the 20 categories and their sizes.

Table 5.2 List of categories and number of documents in the 20 Newsgroups dataset

20 Newsgroups			
Category	Total number of documents	Category	Total number of documents
alt.atheism	799	comp.windows.x	985
comp.graphics	973	misc.forsale	975
comp.os.ms-windows.misc	966	rec.autos	989
comp.sys.ibm.pc.hardware	982	rec.motorcycles	996
comp.sys.mac.hardware	963	rec.sport.baseball	994
rec.sport.hockey	999	soc.religion.christian	996
sci.crypt	991	talk.politics.guns	909
sci.electronics	984	talk.politics.mideast	940
sci.med	990	talk.politics.misc	775
sci.space	987	talk.religion.misc	628

5.2.3 World Wide Knowledge Base (WebKB) Dataset

The WebKB dataset [134] is a collection of 8,282 web pages obtained from four academic domains. WebKB was collected by Craven et al. [135]. The web pages in the WebKB dataset are labelled using two different divisions; the first according to the topic and the second according to the web domain. In the experiment, the first division is considered which consists of seven categories that include “course”, “department”, “faculty”, “project”, “staff”, “student”, and “other”. Following the example of other researchers such as Nigam et al. [136] and Bekkerman et al. [133], the categories “other”, “department”, and “staff” have been discarded in the experiment.

The remaining dataset contains 4,199 documents in four categories. Table 5.3 specifies the four remaining categories and their sizes. This dataset is used in many applications in text mining such as text classification [36, 38] and text clustering [128, 133].

Table 5.3 List of classes and number of documents in the WebKB dataset

WebKB	
Category	Total number of documents
project	504
course	930
faculty	1124
student	1641

5.3 English ATC System Design

To evaluate any text representation model, it must be a part of a particular target application such as an ATC system. The different representation models are evaluated in terms of their effectiveness in an ATC system. The higher classification quality outcome of an ATC system indicates a more effective representation. Three English datasets introduced in Section 5.2 were used to evaluate the effect of the text representation model on the English TC problem.

To build the BOW model, all documents in the dataset are converted to lower case. Then, each document in the dataset is tokenized into words, and each word is stemmed with Porter's stemmer [11]. All stop words are removed using a stop word list that has 524 stop words [137]. In addition, rarer words which appear less than a certain threshold in a dataset are removed which is four in our experiments. This threshold gives the best results in term of classification accuracy and reducing the high dimensions of feature space which reported by other works [85, 138] and confirmed by our experiments. The remaining words are used to

represent documents using the VSM, and each word is weighted using TFIDF (see Section 2.2.2).

For the BOC model, the Wikipedia XML dump files from 2013, which contain 4.2 million articles/concepts are used in this thesis. An open source toolkit is known as Wikipedia-Miner [139] was used to process the dump files and create a database that contains a summarized version of Wikipedia's content and structure. The toolkit assigns each Wikipedia page a unique identifier, title, and type (article, category, or redirect).

First, a document is processed and cleaned. Then, all possible Wikipedia concepts in the document are identified using the topic detection functionality in Wikipedia-Miner. The majority of detected concepts are ambiguous as they may refer to multiple meanings. To disambiguate concepts, the supervised disambiguation algorithm by Milne and Witten [115], which uses bagging [49] of C4.5 decision trees [45] and is trained with 100 randomly sampled Wikipedia articles, is used. The disambiguation computes a probability score for each potential concept based on its relevance to other concepts detected in the document. This probability determines if a concept is relevant to the context or not and the ambiguity is resolved. A set of candidate concepts are then selected which are used to represent documents in the BOC model by the VSM, and each concept is weighted using the TFIDF weighting scheme. Another text representation model is constructed by combining the BOW and the BOC; the combining model enriches the BOW model with concepts from the English Wikipedia site using different strategies, namely AC, RTC, ACC, AUC, and CTD (see Sections 3.6 and 4.2).

Most researchers who used concepts in TC with representation models adopt the SVM algorithm [85, 126] since previous studies found SVMs to have the best performance for TC [8, 27, 140]. Different algorithms have different behaviours that might impact the performance of a representation model. Therefore, SVM, NB, DT and RF as used as classification algorithms and their performances are compared with different representation models.

To measure the performance of the ATC system with different strategies to represent text, different performance measures are used such as the Accuracy, Precision, Recall and F1-measure (See Section 2.3.3).

A ten-fold cross-validation is performed for both the Classic 3 and WebKB datasets, where for the 20 Newsgroups dataset a four-fold cross-validation is performed to compare it to others such as Gabrilovich and Markovitch [85] and Wang et al. [126]. In addition, for all datasets, the hold-out method is used to produce different sizes of training subsets such as 75%, 66%, 50%, 33%, 25%, 16%, and 12%. The percentages refer to the size of subset used for training, with the rest used as a subset for testing. For example, in the case of 75% used for training, 25% is used for testing. These experiments show how the size of the training subset affects the performance of the ATC system.

For both Classic 3 and WebKB datasets, both Micro and Macro- averaged F1-measures are reported, since their category sizes are significantly different. The Micro-averaged F1-measure operates at the document level and is primarily affected by categorisation performance on larger categories. On the other hand, Macro-averaged F1-measures averages result for individual categories and thus small categories with few training examples have a large impact on the overall performance.

5.4 Experiment Result and Evaluation

To analyse the results a comparison between the BOW and BOC and their effect on the English ATC system is demonstrated with different classification algorithms. Then a comparison between the BOW and combined model using different strategies is conducted. The comparison includes the accuracy of the ATC system when these strategies are used with different classification algorithms, the size of the VSM, and the time that it needs to build a classifier.

From Figure 5.1 it can be concluded that the BOW is a better text representation model in the English ATC system than the BOC when used with SVM, NB, and DT classification ML algorithms with all datasets, which is in line with other studies such as [93].

On the other hand, when RF ML algorithms are used in the English ATC system, the BOC model achieved a higher accuracy than the BOW by 3% and 12% with the Classic 3 and 20 Newsgroups datasets respectively.

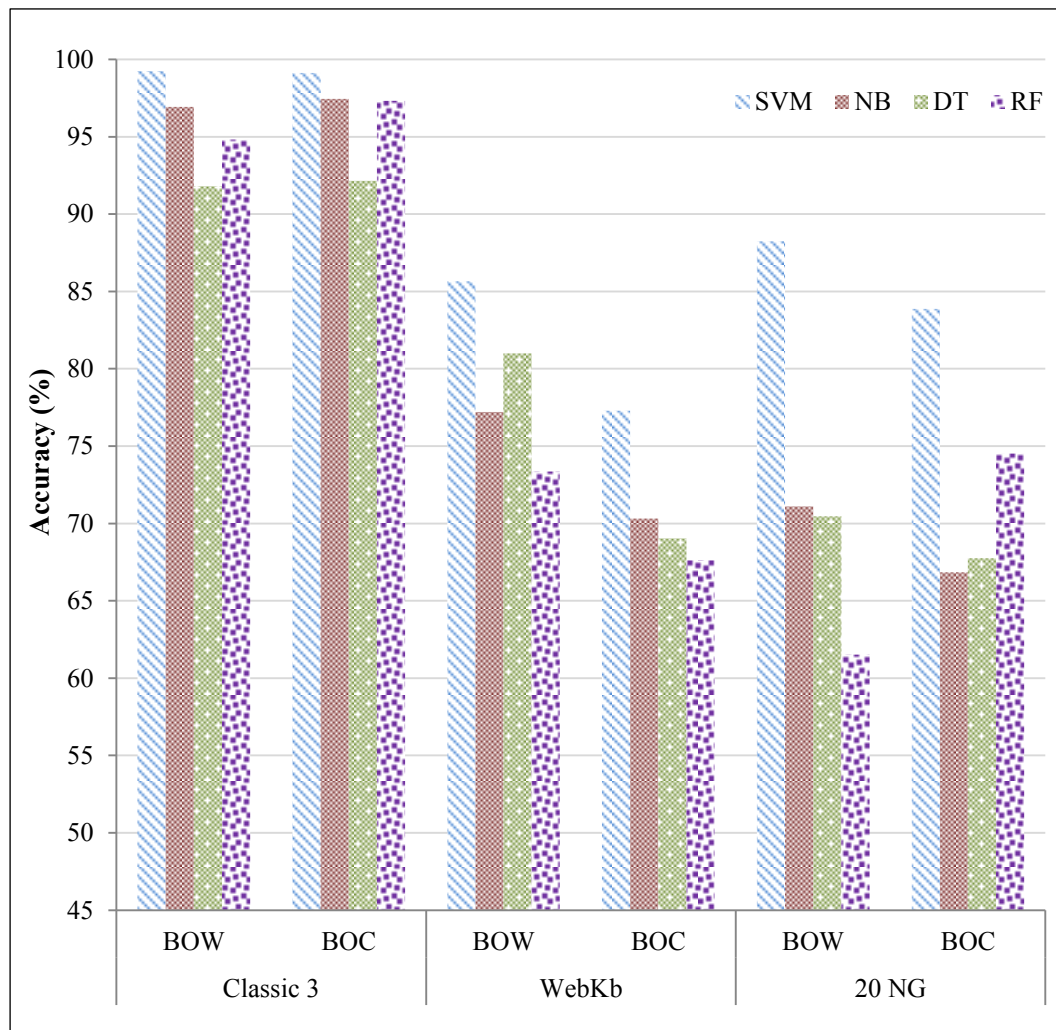


Figure 5.1 Accuracy of the BOW and BOC with different ML algorithms in an English ATC

As can be seen from Figure 5.2, the higher accuracy of the ATC system for Classic 3 dataset with all combined models is achieved by SVM followed by NB, RF, and DT classification algorithms. The SVM as ML algorithm shows its efficiency in many applications in comparison to other ML algorithms.

From Figure 5.2 it can be concluded that the SVM classification algorithm has the best accuracy of the ATC system in comparison to others ML algorithms.

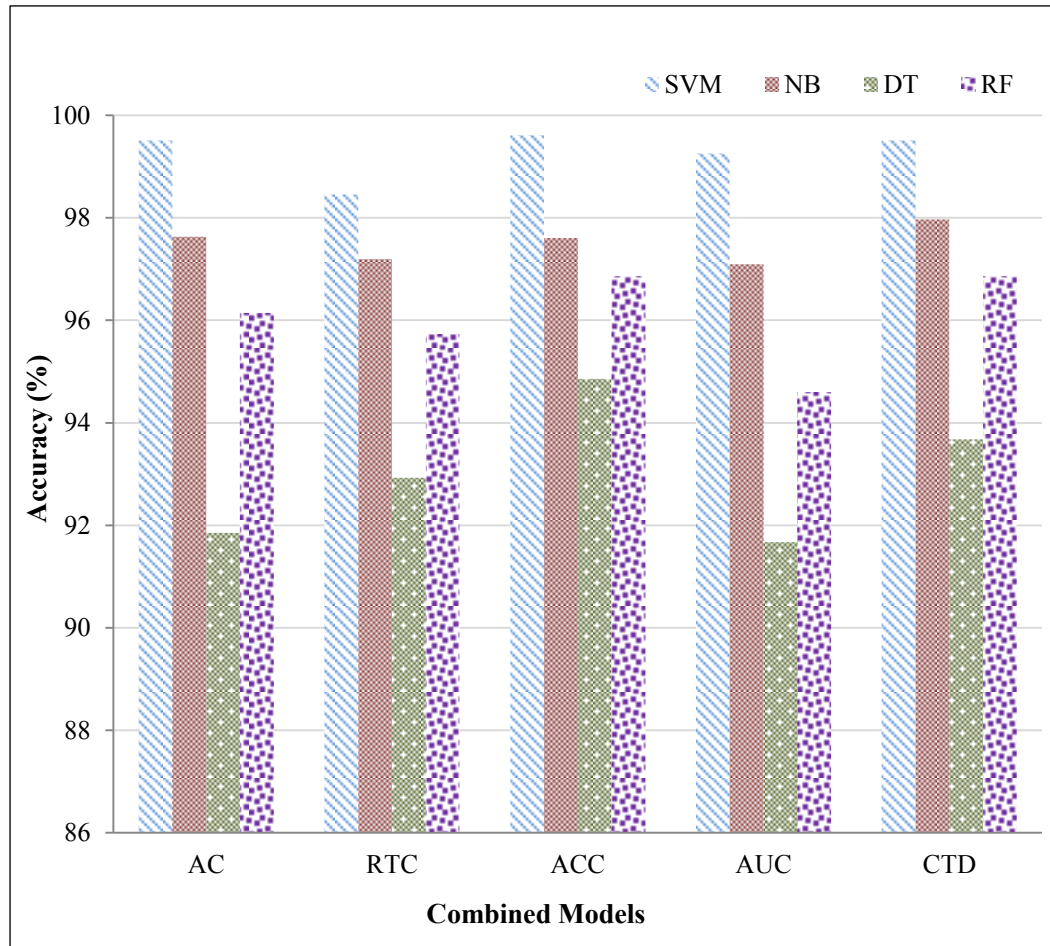


Figure 5.2 Accuracy of combined models with different ML algorithms in the Classic 3 dataset

In order to find the best strategy to build a combined text representation model, the performance of the four classification algorithms is averaged as can be seen in Figure 5.3.

From Figure 5.3 the ACC strategy achieved the best accuracy of 97.23% in comparison to other strategies when it is used to represent documents from the Classic 3 dataset.

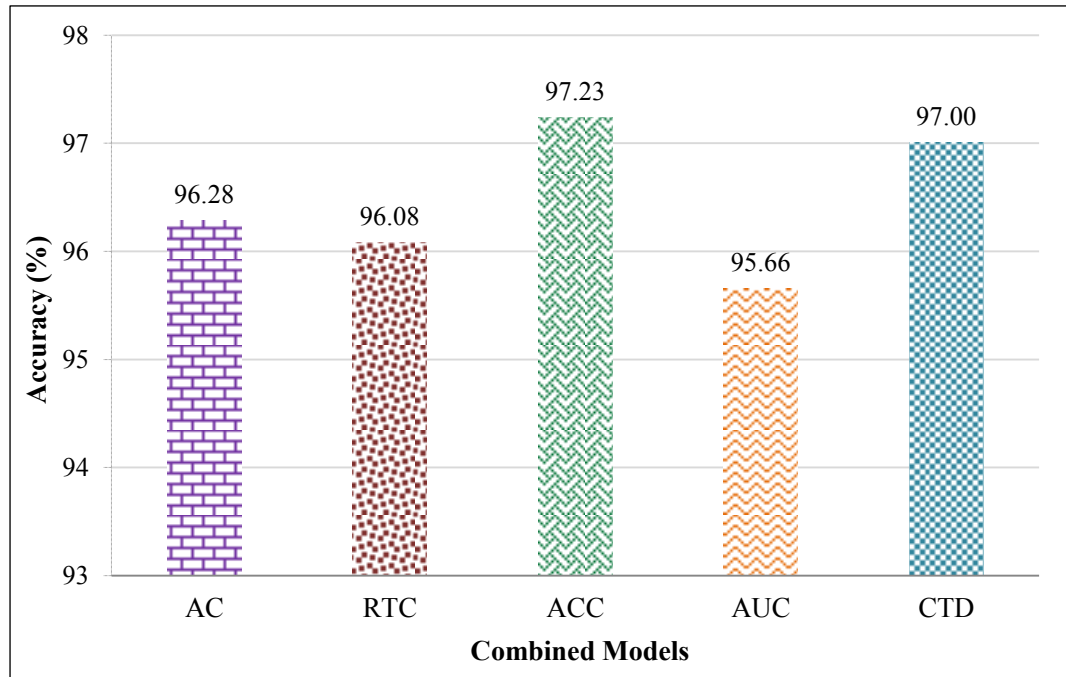


Figure 5.3 Average accuracy with different strategies to build combined models in the Classic 3 dataset

Our developed strategy, CTD, comes second which is only lower than ACC by 0.23. In addition, both the ACC and CTD strategies achieved higher accuracy than the BOW model when used to represent the Classic 3 in the ATC system; see Figure 5.4.

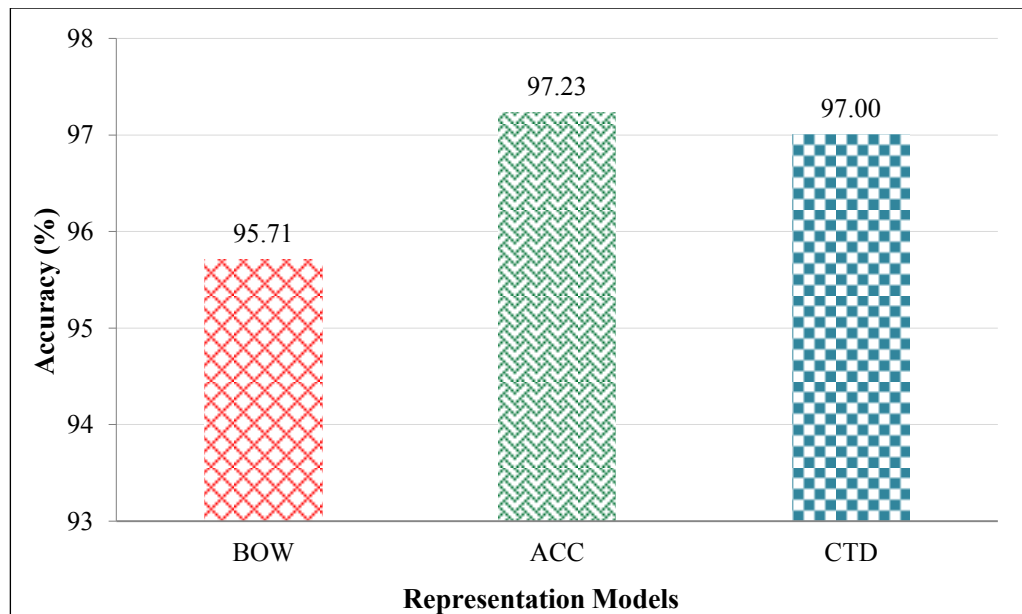


Figure 5.4 A comparison between the BOW and combined models using the ACC and CTD strategies with Classic 3 in the ATC system

The combined model using the CTD strategy achieves a higher accuracy than the BOW by 2.90%, while it has the same number of features when compared to the BOW as seen in Figure 5.5. In addition, the combined model using the CTD strategy uses 39% less features than ACC combined model. However its classification accuracy is only 0.23 lower than that of the ACC.

The largest representation model in term of the number of features is the combined model using the ACC strategy, and the smallest is the combined model using the RTC strategy.

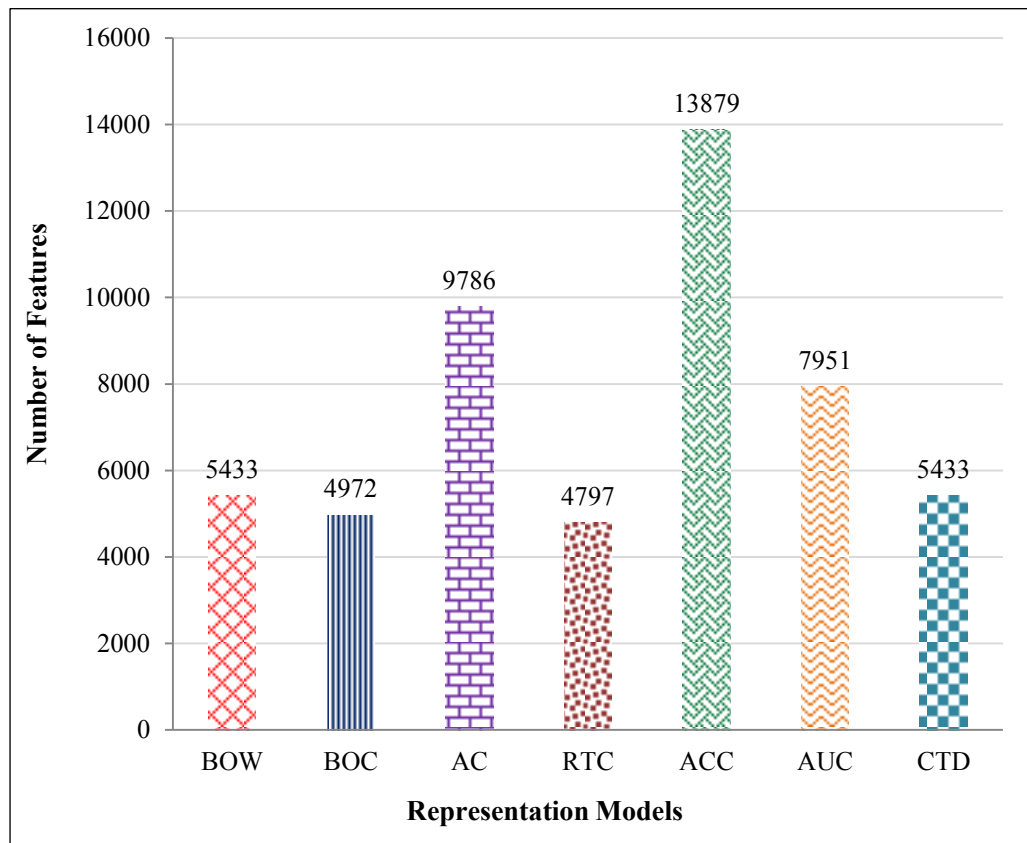


Figure 5.5 Number of features in different representation models in the Classic 3 dataset

To show how the size of a training subset affects the performance of different representation models in the ATC system, the hold-out method is used with different sizes of training subsets such as 75%, 66%, 50%, 33%, 25%, 16% and 12%. The SVM ML classification algorithm is used to build and train a classifier model from the training subset and thereby predict the category of the test subset. Figures 5.6 and 5.7 show that combined models using the ACC, AC, and CTD strategies achieved a higher result than others with both Micro- and Macro-

average F1-measures. In addition, the combined models using the CTD strategy perform better than the combined models using the AC strategy up to the point when 16% of the dataset is used as a training subset. Then combined model using the AC strategy do better than using the CTD strategy.

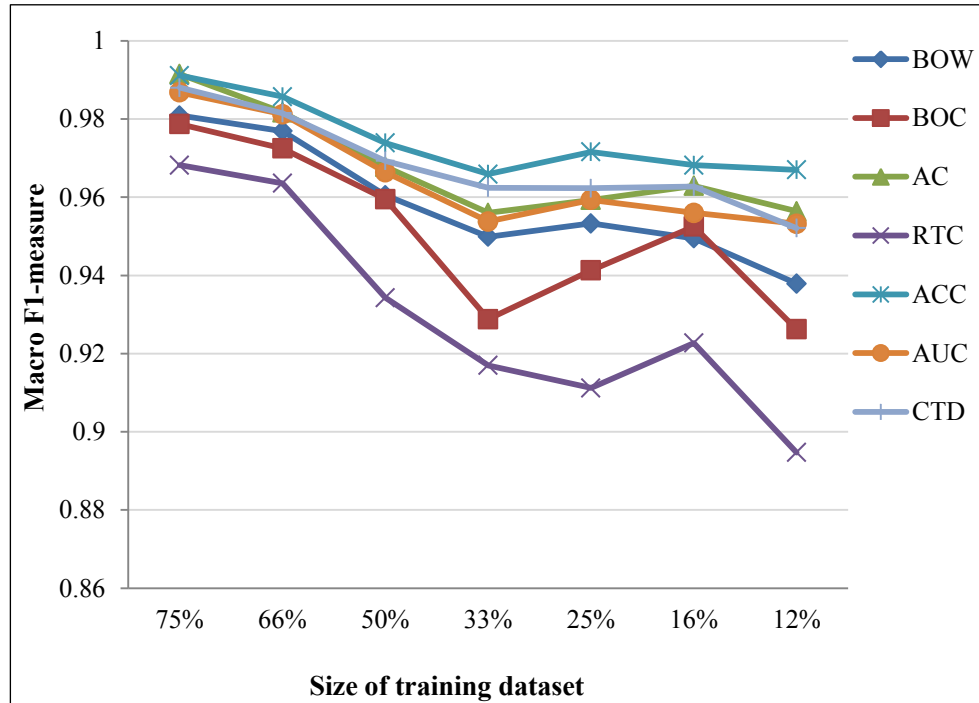


Figure 5.6 Macro F1-measure of different representation models with different size of training subset of Classic 3 dataset with SVM classifier

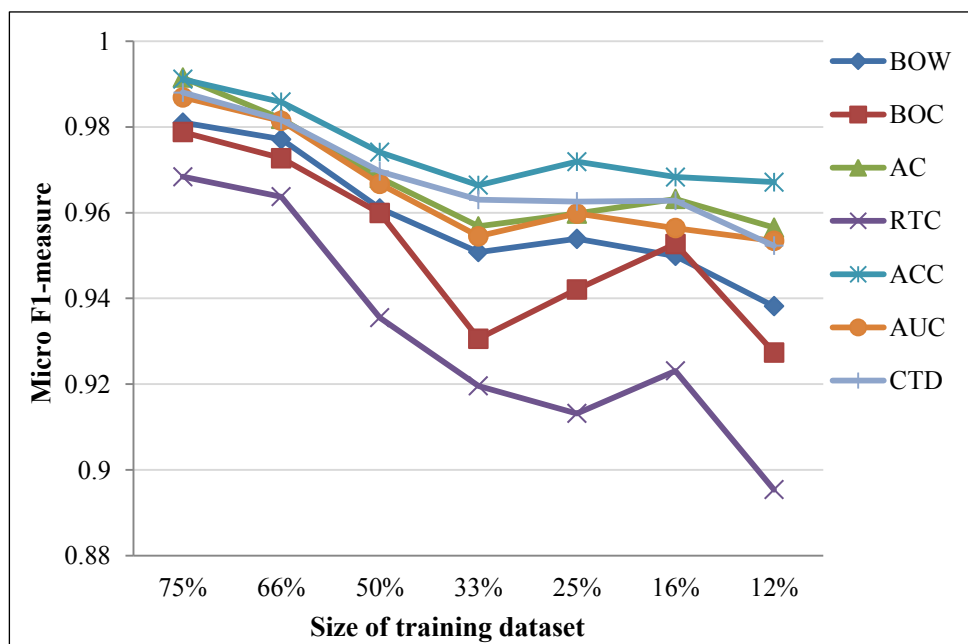


Figure 5.7 Micro F1-measure of different representation models with different size of training subset of Classic 3 dataset with SVM classifier

The WebKB dataset is also used to test different representation models in the English ATC system. Web pages included in this dataset are not evenly distributed across categories; the “student” category has 1,641 web pages while the “project” category only has 504 web pages. Therefore, the Macro F1-measure is reported to find the average results for individual categories, so small categories with less training examples have an equal impact on the overall performance, whereas reporting the Micro F1-measure is primarily affected by categorisation performance on larger categories as it operates at the document level.

As mentioned earlier, it can be concluded from Figure 5.1 that the BOW is a better text representation model in the English ATC system than the BOC. Therefore, a comparison between the BOW model and combined models using different strategies is needed to represent the WebKB dataset in the English ATC system.

As can be seen from Table 5.4, the best performance is achieved by the combined model constructed with the AUC strategy which achieved an accuracy of 86.21%. In addition, the BOW model achieved a higher accuracy than other combined representations which are AC, RTC and CTD. This is due to the fact that the BOW representation has more distinguishing features between categories in the WebKB dataset than in any other representation model.

The webpages in these categories share a lot of features, for example, a webpage from the “project” category discusses a project about graphics and image processing. It shows the name of the laboratory of this project and some related courses, publications, webpages, journals, and conferences. Where a webpage from the category “faculty” contains information about a lecturer such as courses, laboratories, publications, and office hours.

By using concepts to represent the WebKB dataset, we get fewer features in comparison to words and most of these features are general and repeated between categories. Therefore, any strategy that uses concepts to build the combined representation model suffers from low performance when it classifies text from the WebKB dataset.

Table 5.4 The performance of the SVM classifier with combined models and the BOW model to represent text in the WebKB dataset

	Accuracy (%)	Micro F1- measure	Macro F1- measure	Average Recall	Average Precision
BOW	85.66	0.857	0.858	0.857	0.857
AC	85.26	0.853	0.859	0.853	0.853
RTC	79.76	0.797	0.798	0.798	0.797
ACC	85.85	0.859	0.862	0.859	0.859
AUC	86.21	0.862	0.854	0.862	0.862
CTD	84.76	0.848	0.837	0.848	0.848

As can be seen in Table 5.5 that the combined model which used the AC strategy achieved the best classification accuracy of 77.66% when the NB classifier is used. In addition, both BOW and AUC representation models achieve a high accuracy of 77.21% and 76.54% respectively, as can be seen in Table 5.5.

Table 5.5 The performance of the NB classifier with combined models and the BOW to represent text in the WebKB dataset

	Accuracy (%)	Micro F1- measure	Macro F1- measure	Average Recall	Average Precision
BOW	77.21	0.770	0.745	0.772	0.778
AC	77.66	0.774	0.757	0.777	0.779
RTC	73.61	0.735	0.720	0.736	0.741
ACC	70.30	0.703	0.755	0.703	0.724
AUC	76.54	0.763	0.755	0.765	0.772
CTD	76.40	0.763	0.724	0.764	0.769

By using the DT classifier, as can be seen in Table 5.6, the BOW representation model achieved the best classification accuracy of 80.99%. In addition, the combined model which used the AUC strategy has a better accuracy than other combined models achieving an accuracy of 80.90%.

In conclusion, both BOW and AUC representation models achieve a high accuracy when both the NB and the DT are used as classification algorithms in the ATC, as can be seen in Tables 5.5 and 5.6.

Table 5.6 The performance of the DT classifier with combined models and the BOW to represent text in the WebKB dataset

	Accuracy (%)	Micro F1- measure	Macro F1- measure	Average Recall	Average Precision
BOW	80.99	0.803	0.776	0.810	0.809
AC	80.23	0.793	0.767	0.802	0.800
RTC	75.83	0.766	0.734	0.758	0.755
ACC	80.33	0.802	0.781	0.803	0.802
AUC	80.90	0.808	0.7956	0.809	0.808
CTD	79.23	0.792	0.782	0.793	0.791

When the RF is used as a classifier, the best classification accuracy is achieved with the CTD strategy when it is used to build the combined model which achieved an accuracy of 74.86 %. In addition, both BOW and AUC representation models achieve a high accuracy of 73.04 % and 76.54 % respectively, as can be seen in Table 5.7.

Table 5.7 The performance of the RF classifier with combined models and the BOW to represent text in the WebKB dataset

	Accuracy (%)	Micro F1- measure	Macro F1- measure	Average Recall	Average Precision
BOW	73.04	0.717	0.673	0.730	0.726
AC	71.83	0.704	0.664	0.718	0.714
RTC	71.64	0.700	0.653	0.716	0.714
ACC	72.28	0.707	0.658	0.723	0.714
AUC	73.54	0.724	0.686	0.735	0.733
CTD	74.86	0.736	0.689	0.749	0.746

From the Tables 5.4-5.7, one can see the Micro-averaged F1-measure of the ATC system and how the combined model constructed using the AUC strategy was not affected by the size of the categories in the WebKB dataset.

As seen earlier, representation models are affected by the different sizes of categories in the WebKB dataset where some categories have fewer web pages than others and the classifier does not get enough training examples for small categories.

To compare different representation models when different sizes of training subsets are used and how the performance of the ATC system is affected by the different size of the representation model the hold-out method is used with different sizes of training subsets such as 75%, 66%, 50%, 33%, 25%, 16%, 12%

and 10%. The SVM ML classification algorithm is used to build and train a classifier model from the training subset and thereby predict the category of the test subset. As can be seen from Figure 5.8, the combined models using the ACC, CTD, and AUC strategies achieved a higher Micro-averaged F1-measure than other models. In addition, combined models using the CTD strategy achieved higher classification accuracy than all models when only 25% of the dataset is used. On the other hand, when only 10% of the dataset is used as a training subset the CTD strategy performed like the BOW, which is lower than combined models using the strategies ACC, RUC, and AC.

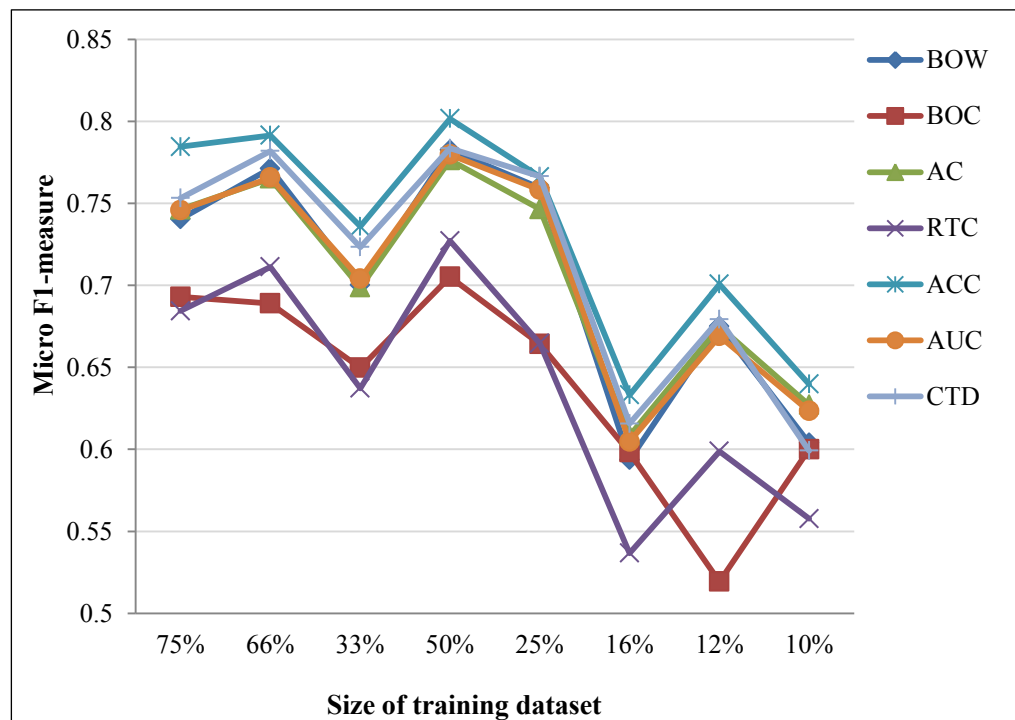


Figure 5.8 The Micro F1-measure of different representation models with a different size of training subset of the WebKB dataset to train the SVM classifier

As indicated earlier for the 20 Newsgroups dataset, the BOW model achieved a higher classification with the SVM, NB, and DT than the BOC model as can be seen from Figure 5.1, where the BOC model, in conjunction with the RF classifier, achieved a higher result than the BOW model.

In Figure 5.9 a comparison between all representation models for the 20 Newsgroups dataset can be seen. The SVM ML algorithm is used, and the best performance is achieved by the ACC with an average Micro F1-measure of 0.90 and an average Macro F1-measure 0.896. This is then followed by the CTD with a

0.8967% average Micro and 0.893 average Macro F1-measure where the AC has an average Micro F1-measure of 0.896 and a Macro F1-measure of 0.891.

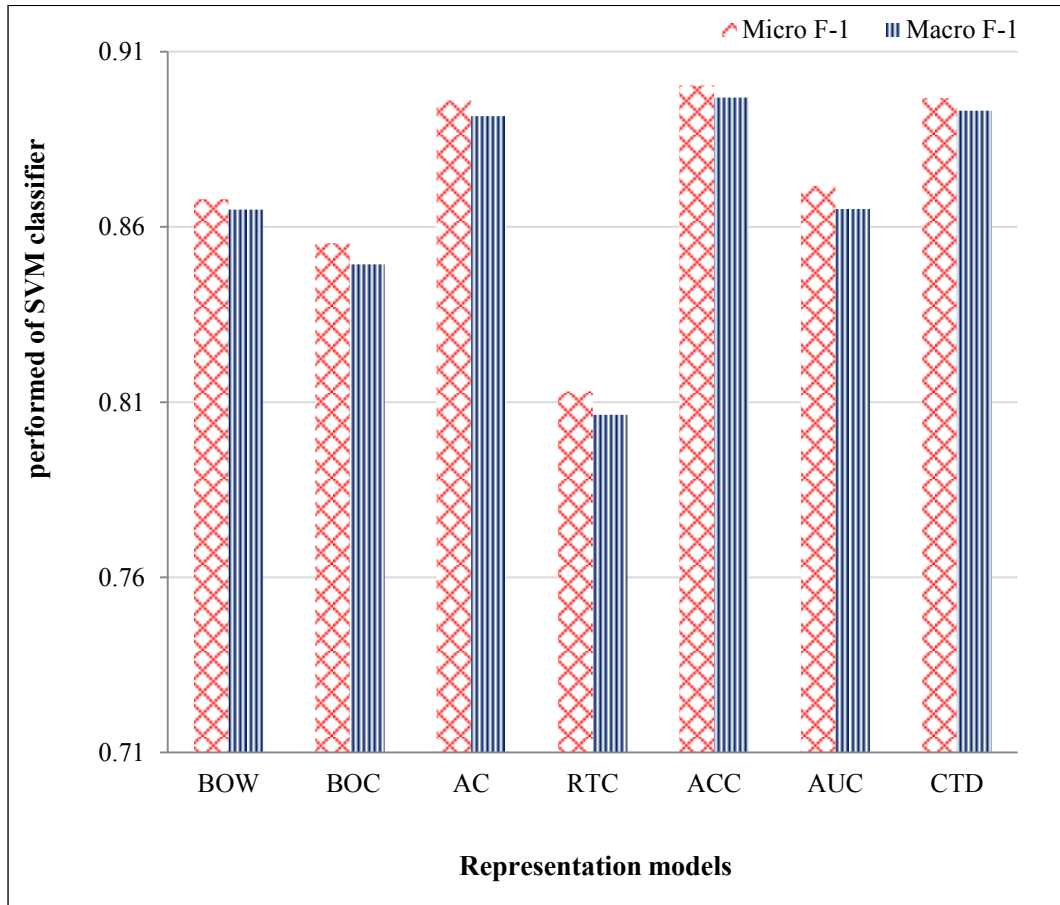


Figure 5.9 Performance of the SVM classifier with different representation models for the 20 Newsgroups

The 20 Newsgroups dataset is one of the most popular datasets used to evaluate the performance of an ATC system. Many researchers use it in their work as benchmark datasets to compare their combined text representation models with the standard BOW model. For example, Gabrilovich and Markovitch [85] developed a strategy that built a combined text representation model that achieved an average Micro F1-measure of 0.862%, with a 1% improvement more than the BOW baseline. Wang et al. [126] developed a strategy that added concepts, parent categories for these concepts, and top 5 related concepts to the basic BOW model. It achieved an average Micro F1-measure of 0.917% and an average Macro F1-measure of 0.904%.

Table 5.8 shows the different studies that used the 20 Newsgroups dataset and the average Micro F1- measures of their combined model in ATC system. The best

classification accuracy is achieved by the combined model developed by Wang et al. [126] with an average Micro F1-measure of 0.917%. Our developed CTD strategy used to enrich the text representation model by embedding concepts weight in the standard BOW model achieved an average Macro F1-measure of 0.893%.

Table 5.8 Average Macro F1-measure for 20 Newsgroups in different combined models using different strategies

Combined model	Features in VSM	averaged Micro F1
AC [85]	Words, concepts	86.2%
combined model (Wang et al. [126])	Words, concepts, concepts categories, top 5 related concepts	91.7%
ACC	Words, concepts, concepts categories,	90%
AC	Words, concepts	89.67%
CTD	Words	89.61%

The CTD combined model achieved an improvement of 3.12% over the combined model by Gabrilovich and Markovitch [85]. On the other hand, the CTD combined model achieved less with 2.07% than the combined model by Wang et al. [126] which contained words, concepts, concepts categories, and top 5 related concepts. The CTD combined model has only words as features and concepts weight is embedded in the words. Therefore, the CTD combined model has fewer features in comparison to other combined models and has the same number of features as the standard BOW model. As can be seen in Figure 5.10, the largest representation model is constructed using the ACC strategy to build the combined model with

63922 features where the smallest is RTC with only 17,107 features. The RTC combined model contained about 26% the size of the ACC combined model where our developed CTD combined model contained about 29% the size of the ACC model. In addition, the CTD combined model achieved classification accuracy lower by only 0.3 than ACC combined model.

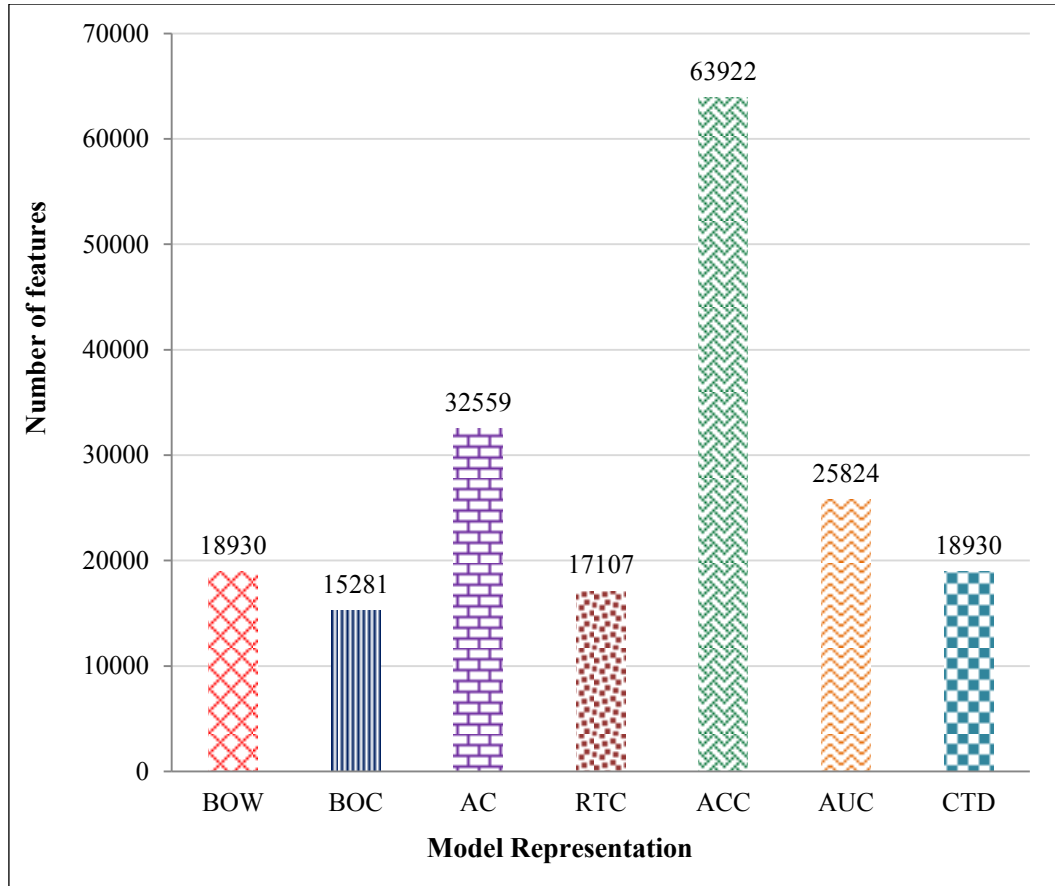


Figure 5.10 Number of features in different representation models in 20 Newsgroups

From Figure 5.11 it can be seen that with the NB classifier all representation models achieved the same level of classification accuracy. The best performance was achieved with the AC combined model with an average Macro F1-measure of 0.735. Our developed CTD combined model comes next with an average Macro F1-measure of 0.725. The best performance of representation models is achieved by the standard BOC model when the RF classifier is used. The combined models ACC and RTC come next with only 1% lower than the BOC model.

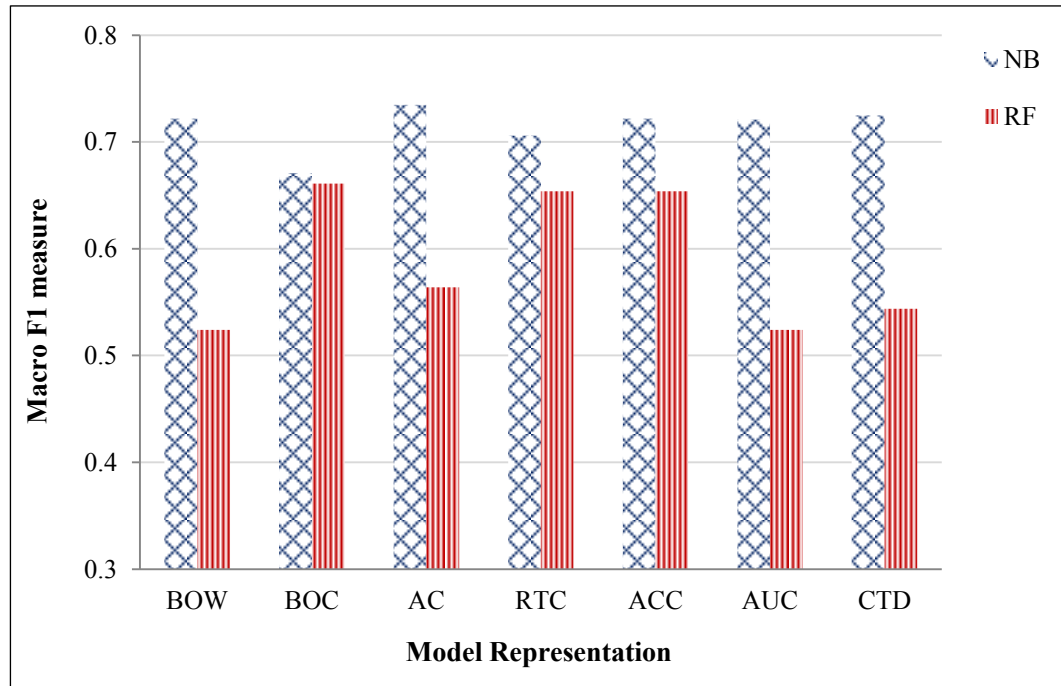


Figure 5.11 Performance of NB and RF classifier with different representation models for 20 Newsgroups

5.5 Discussion of Results

The aim of experiments in this chapter is to show how text representation models affect the performance of an English ATC system. To achieve this, concepts and words as features are used in the experiments. These features are used separately by creating the BOW (see Section 3.2) and BOC (see Section 3.5) models and are combined by using different strategies such as AC, RTC, ACC, AUC, and CTD (see Section 3.6, and 4.2). Wikipedia as a KB is used to represent text in the BOC.

Different ML algorithms such as SVM, NB, DT, and RF are used to classify documents in different datasets. In the previous section (see Section 5.4) the result of the classification accuracy is analysed for each dataset separately with different text representation and classification algorithms whereas in this section the classification accuracy results of three datasets are averaged to discuss the results.

By comparing the average accuracy of the BOW and BOC models in all datasets with different ML algorithms, as can be seen in Figure 5.12, it can be concluded that the BOW works better than the BOC in the English ATC system with SVM, NB, and DT, whereas by using RF ML algorithms, the BOC gives a higher accuracy than the BOW by 3.25%.

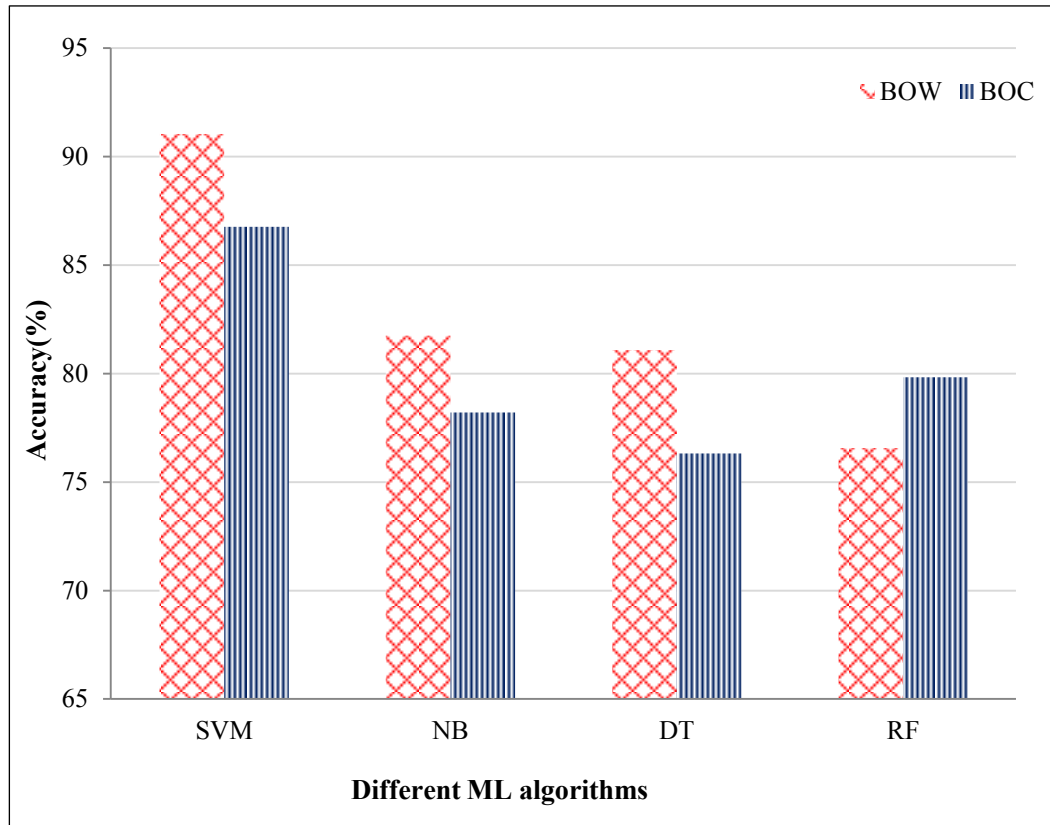


Figure 5.12 Average accuracy of different classifiers with BOW and BOC models in all datasets

The RF ML algorithm manages concepts as features better than words and as a result, it gives a higher classification accuracy for the BOC model. The RF selects random subsets of training data to build a random tree which is based on selecting a random subset of features (see Section 2.2.3.5).

By averaging the performance of different representation models in the three datasets with different classifiers, as can be seen in Figure 5.13, the ACC and AC models achieved higher average Macro F1-measures of 82.92% and 82.9% respectively.

The CTD, we developed and discussed (see Section 4.2) achieved an average Macro F1-measure of 82.30% which is less than the ACC and AC by 0.53 and 0.51 respectively. In addition, the performance of CTD is better than the BOW by 0.53 and BOC by 1.8% as can be seen in Figure 5.13.

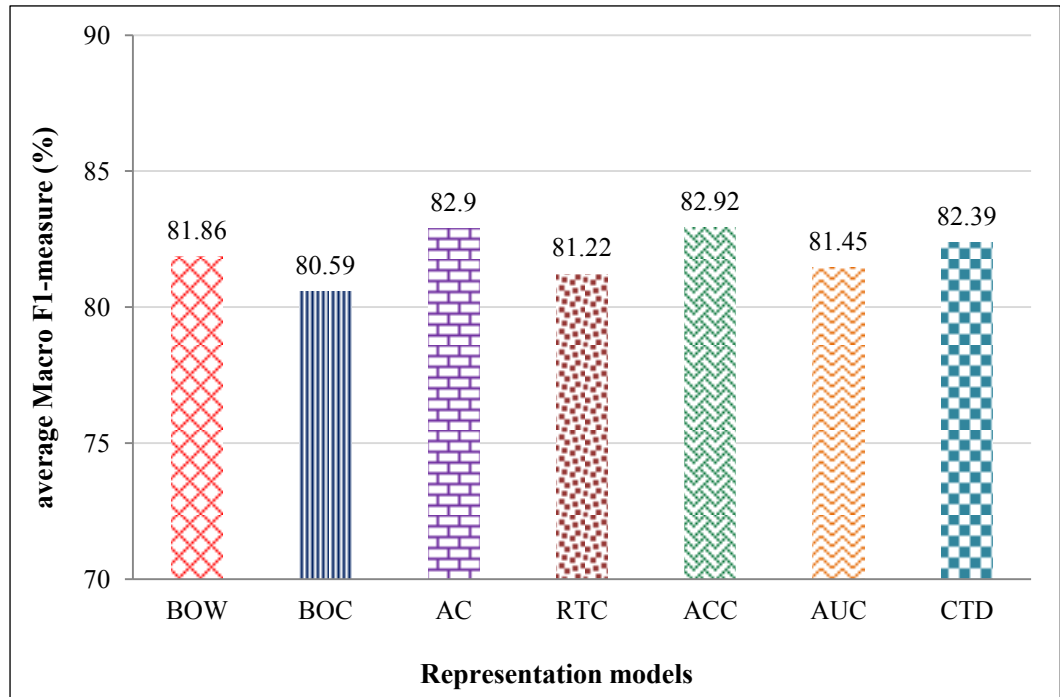


Figure 5.13 Average Macro F1-measure of different representation models on all datasets

Furthermore, the developed CTD model uses a minimum number of features in comparison to other combined models such as ACC, AC, and AUC. The CTD model has the same number of features of the BOW model, and it only has 31% of the ACC size and 60% of the AC size as can be seen from Figure 5.14.

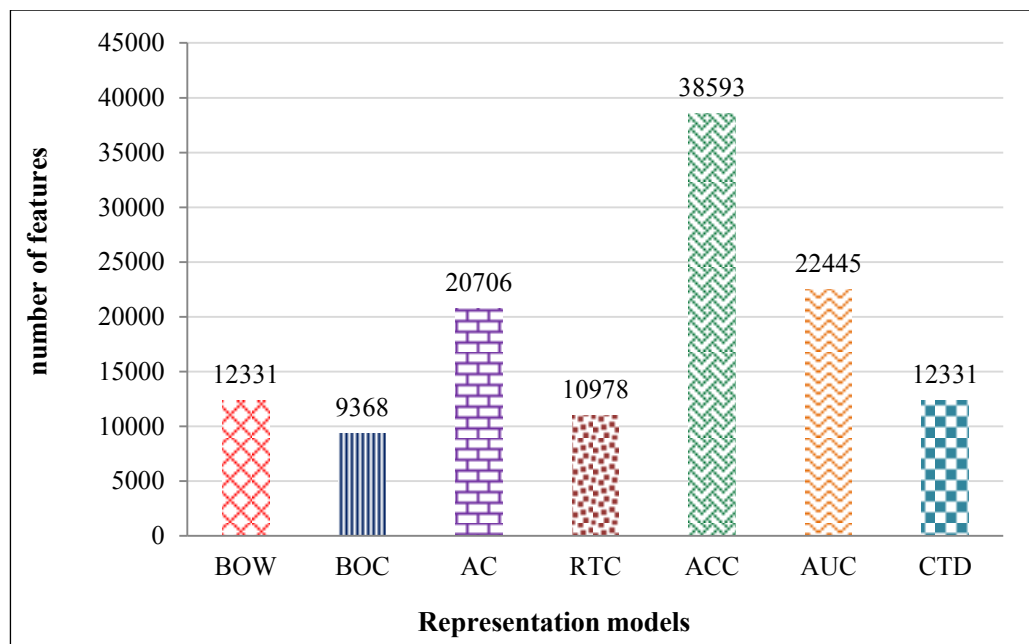


Figure 5.14 Average number of features in different representation models for all datasets

5.6 Summary

From this chapter, it can be seen that the developed strategies show promising results when used to represent English text in ATC systems. The experiment design is described, and the results of the experiment with proposed strategies are compared to the results of other strategies. In addition, the hold-out and cross-validation approaches used in these experiments show how the size of the training subset affects the representation models. The experiment result demonstrated the efficiency of different capabilities of these models when applied to the English ATC systems as compared to the BOW alone and BOC model. The ACC achieved the best performance followed by our developed strategy CTD, which has been shown to achieve higher categorisation accuracy while using the minimum number of features. Furthermore, when SVM, NB, and DT ML algorithms are used in the ATC system, the standard BOW model and the combined models achieved higher classification accuracy than the BOC model. On the other hand, using the BOC model as a representation for English text in ATC system in conjunction with the RF ML algorithm showed better efficiency.

CHAPTER 6

Arabic Text Classification using Combined Representation

6.1 Introduction

In this chapter, the Arabic language is discussed in relation to its complexity and how it compares to other languages such as English and the impact this has on the experiments in this thesis. The Arabic datasets used in the experiments to evaluate the Arabic ATC system are outlined and discussed in detail. Various studies that used Arabic text in an ATC system are reviewed in relation to the diverse components of ATC systems. A discussion of the experiment design and results reveals how text representation models affect the accuracy of the Arabic ATC system. We then compare and contrast the strategies developed in this thesis with other established strategies. In addition, the impact of the morphological analysis is investigated in regards to the classification accuracy of the Arabic ATC system, the size of the representation models, and the time required by ML algorithms to process the training data and build a model to classify test data.

6.2 Main Features of the Arabic Language

Arabic is the fifth-most popular spoken language in the world. It is spoken by 186 to 422 million people as their first language and by 250 million people as a second language [141].

The Arabic alphabet consists of 28 letters (أ ب ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ) (ي و ه ن م ل ك ق ف) and the hamza (ء). Researchers have different opinions about hamza (ء), a group of researchers considered hamza (ء) as an extend for the letter alif (ا) which is used to represent a glottal stop of the alif (ا) [142]. On the other hand, other researchers considered it as a full letter which is independent of an alif (ا) but is not a part of the alphabet, and that is why hamza (ء) is separated it from the rest of the alphabet [143]. Unlike English, Arabic text is read from right to left. There is no upper or lower case for Arabic letters, and proper nouns are not capitalised which makes the process of recognizing and extracting these nouns difficult for the ATC system.

To form a word, Arabic letters are directly connected, and 22 of the letters take different shapes based on their position in the word be it initial, medial, or final. Table 6.1 shows the letter “b” (ب) appearing in different sample words and how its position affects its shape.

Table 6.1 Example of an Arabic letter “b” (ب) and its various shapes depending on its position in the word

Position	Word	Letter shape
Start	بدر	ب
Middle	قبر	ـبـ
End	قلب	ب

The Arabic language belongs to the Semitic group of languages, and it has a rich morphology with a large degree of inflection and derivation [144], which makes pre-processing a text a challenging task in an Arabic ATC system.

Arabic words can be divided into three main groups; nouns, verbs and participles. A noun in Arabic is defined as a word that describes a person, thing, place, or idea and it can be derived from other nouns, verbs, or participles [145]. Verbs in Arabic are divided into perfect, imperfect, and imperative cases. In Arabic, participle categories include pronouns, adjectives, adverbs, conjunctions, prepositions, interjections, and interrogatives [145].

All words in Arabic must have a gender be it male (مذكر) or female (مؤنث) [146]. This is a significant feature of Arabic, unlike English where this distinction is not obvious for the majority of words. For example, in English “teacher” does not suggest the gender of the person it refers to, while in Arabic (مدرس) is a male teacher and (مدرسة) is a female teacher. In Arabic, two words could refer to the same object and carry different genders, for example, there are two words that mean “window”; (شباك) which is a male window and (نافذة) which is a female window. Gender is not only specified in the nouns, it must also be designated in all grammatical elements that attend them in a sentence such as adjectives, relative, and personal pronouns [146, 147].

In addition to the gender distinction, Arabic has singular, dual, and plural forms of pronouns, nouns, verbs, and adjectives [146]. The singular form is used when referring to one person or thing, the dual subject pronoun to refer to two people or

two things, and the plural form when referring to more than two people or things [146, 147].

In Arabic, the roots of words are composed of three, four, or in some cases, five letters [148]. To obtain Arabic words from stem or root, an affix such as a prefix, suffix, or infix is attached in or to the root or stem [149, 150].

Prefixes in Arabic are groups of letters which are added to the beginning of the word; these letters could be definite articles, prepositions, pronouns, or conjunctions. For example, as it can be seen in Table 6.2 (ال) is a prefixes that is added to the beginning of the word (كتاب) “book” to be defined as “a book” (الكتاب).

Suffixes in Arabic are sets of letters which are added to the end of the word and indicate possessive pronouns. Table 6.2 shows an example of the pronoun (ها) which means “her” such as in “her book” (كتابها). Infixes in Arabic are sets of letters that are inserted into the root to create new words. For the root (كتب) “write”, the word “writer” (كاتب) can be created by inserting (ل) in the second position of the root as seen in Table 6.2. The complex nature of stemming Arabic words has leded researchers to develop different methods to stem a word in Arabic ATC systems (see Section 6.3).

Table 6.2 List of string letters that represent prefix, suffixes, and infixes in Arabic

Affix	String of letters
Prefix	(ال) and (فال), (كال), (بال), (وال)
Infix	(ي) and (ل), (ت), (و), (ن)
Suffix	(ي) and (يه), (ات), (ها), (ن ا), (ن و), (يه)

While there is a set of phonetic symbols for English to aid with pronunciation, there is no such system of phonetics for the Arabic language. Instead, we must rely upon the set of diacritics present in certain forms of Arabic text to provide information about pronunciation and meaning. In Arabic, each letter could be pronounced in a different way based on diacritic that it has. These diacritics also represent short vowels with marks above or below the letters. These diacritics include fatha, kasra, damma, sukūn, shadda, and tanwin [148, 151, 152]. Table 6.3 shows the letter “b” (ب) appearing with different diacritics.

Table 6.3 Example of Arabic diacritics

Diacritic	fatha	kasra	damma	sukūn	shadda	tanwin
Letter	بَ	بِ	بُ	بْ	بّ	بً

Text in the Arabic language has three forms; Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectal Arabic (DA) [153-155]. CA includes classical historical ritual texts such as religious books, ancient poems, and old literature. In CA, each letter in the text has diacritics whereas MSA is written without these diacritics and is commonly found in newspapers, news media, formal speech, and books. DA refers to the predominantly spoken vernaculars (slang accents) and has no written standards; it differs between Arabic regions and can be found on social media sites such as Facebook, Twitter and blogs.

With the absence of diacritics in MSA texts, learning the language is more difficult for non-native speakers and leads to more challenges for the automatic pre-processing of Arabic text in an ATC system. Multiple ambiguities in the meaning of words causes problems, for example, words such as (ذهب) could be a noun mean “gold” (ذَهَبَ) with fatha on the first and second letters (ذَهْ) or the verb “went” (ذَهَبَ) with fatha on the second and third letter (ذَهَبْ). Another example shows that (كتب) without diacritics could be the plural noun “books” or an active past tense verb “wrote”. Furthermore, in some cases, it is impossible to determine the part of speech of the words without contextual clues. For example, a word like (من) could be a preposition meaning “from”, a question meaning “who”, or a verb meaning “granted”. While ambiguity is a challenge in any language, it can be complex in Arabic.

In comparison with other languages such as English, the difficulty of Arabic Natural Language Processing (NLP) in general and the Arabic ATC system in particular make it a challenging task for the following reasons:

- The Arabic language has more complex morphologies, semantics, and syntactics in comparison with English and other European languages [156]. Table 6.4 shows some of the complex morphology for Arabic and the large degree of inflection and derivation that it has.

Table 6.4 Different forms that the word (كتب) “write” can take in Arabic

Word	Tense	Case	Gender	Meaning
كتب	Past	Singular	Masculine	He wrote
كتبت	Past	Singular	Feminine	She wrote
يكتب	Present	Singular	Masculine	He writes
تكتب	Present	Singular	Feminine	She writes
سيكتب	Future	Single	Masculine	He will write
ستكتب	Future	Single	Feminine	She will write
كتبوا	Past	Dual	Masculine	They wrote
كتبتا	Past	Dual	Feminine	They wrote
يكتبان	Present	Dual	Masculine	They write
تكتبان	Present	Dual	Feminine	They write
سيكتبان	Future	Dual	Masculine	They will write
ستكتبان	Future	Dual	Feminine	They will write
كتبوا	Past	Plural	Masculine	They wrote
كتبن	Past	Plural	Feminine	They wrote
يكتبون	Present	Plural	Masculine	They write
يكتبن	Present	Plural	Feminine	They write
سيكتبون	Future	Plural	Masculine	They will write
سيكتبن	Future	Plural	Feminine	They will write

- There is more than one encoding system for Arabic and the most popular are UTF8, Unicode ISO-8859, and CP-1256 (Windows-1256). In the case of using an inapplicable encoding, it will result in an inappropriate Arabic text display which affects the accuracy of the Arabic ATC system [151].
- In the Arabic ATC system, most of the literature uses small in-house collected datasets or public Arabic datasets but with a small quantity of documents and categories. In addition, the training and testing portions for Arabic datasets used in these works is not publically available. This makes comparisons between Arabic ATC systems difficult [151, 156]. To overcome this issue, three datasets that are frequently used in the literature (two large public access and one small available on request) are used and our results are compared with those from other researchers. We used a ten-fold cross-validation method which included all documents in the dataset for training and testing.

6.3 Experimental Datasets

In this section, three independent Arabic datasets have been used to evaluate the proposed text representation models in an Arabic ATC system. The datasets are diverse in the number of categories and documents which show how the nature of the dataset can affect the accuracy of the ATC system. Full details are provided below for each dataset.

6.3.1 Al-Khaleej Dataset

This dataset is a collection of 5690 Arabic news articles gathered from the online newspaper Al-Khaleej by Abbas and Smaili [157], and it is publicly available [158]. It consists of four categories including international news, local news, sports, and economics. Each document in this dataset has only one category label.

Table 6.5 shows the distribution of the dataset documents among the four categories. The categories vary in size where “Local news” has almost 50% of the dataset while the “Sports” category has 20%.

A number of researchers used this dataset for Arabic text mining such as topic identification [157], text classification [159], and text similarity [160].

Table 6.5 Number of documents per category in Al-Khaleej dataset

Al-Khaleej	
Category	Total number of documents
International news	953
Local news	2398
Sports	1430
Economics	909

6.3.2 Saudi News Paper (SNP) Dataset

This dataset consisted of 5121 Arabic documents from Saudi Newspapers (SNP) and was compiled by Al-Harbi et al. [161]. These articles are of different lengths and belong to seven categories that include culture, economics, general, information technology, politics, social, and sports which are specified in Table 6.6. This dataset is used for the Arabic ATC such as in [161, 162] and for comparison with different stemming methods as in Al-Shammari [163].

Table 6.6 Number of documents per category in SNP dataset

SNP	
Category	Total number of documents
Culture	738
Economics	739
General	728
Information Technology	728
Politics	726
Social	731
Sports	731

6.3.3 Arabic 1455 Dataset

This dataset is collected by Mesleh [138] from online Arabic newspaper archives including *Al-Jazeera*, *Al-Nahar*, *Al-Hayat*, *Al-Ahram*, and *Al-Dostor*. The dataset contains 1445 documents that vary in length and fall into nine categories including computers, economics, education, engineering, law, medicine, politics, religion, and sports. Table 6.7 shows the nine categories and their sizes. The two categories “computers” and “education” have a smaller number of documents compared to the other categories. This dataset is used in Arabic ATC systems such as in [60, 138, 164, 165].

Table 6.7 Number of documents per category in Arabic 1445 dataset

Arabic 1445	
Category	Total number of documents
Computers	70
Economics	220
Education	68
Engineering	115
Law	97
Medicine	232
Politics	184
Religion	227

6.4 Arabic Text Classification

As described in Section 2.1, working with ATC systems for English text started in the 1960s and Maron's work was the first in the field [9]. Since then, many researchers have worked with English text for ATC and with European languages such as French, German, and Spanish [54, 166-169]; others worked with Asian languages such as Chinese and Japanese [170-175].

However, limited work has been done in the Arabic language for ATC in comparison with what has been done in English. Even though several studies have been reported, working with Arabic text in ATC is still in its preliminary stage. Most of the studies focus on comparing different stemming methods, investigating the impact of pre-processing, applying different classification algorithms and comparing them on Arabic text in ATC. In addition, most of the datasets used in these works are in-house datasets, and where there are public datasets available it is limiting to use the same public dataset in more than one work under similar conditions. In the following, most of the common works in Arabic ATC are reviewed from this perspective.

Researchers such as Harrag et al. [176] compared different stemming methods in ATC for Arabic text. The authors showed a comparison between three stemming methods such as Light Stemming (LS), Root Extraction (RE), [177] and dictionary-lookup [178]. The RE method works by removing the suffixes and prefixes attached to a given word and word pattern matching is used to extract the root. The LS method [179] only involves removing a small set of prefixes and suffixes. The LS does not deal with infixes or recognize patterns to find roots. The dictionary-lookup method uses dataset statistics to generate possible roots for a given word and estimates the probability of deriving the word from each of the possible roots. The authors in [176] used the stemming methods to reduce the feature space of the VSM for two different classifiers, the Artificial Neural Networks (ANN) and the Support Vector Machine (SVM). To evaluate the ATC system, an in-house Arabic dataset is used which includes 453 documents distributed over 14 categories. The documents are a set of the Nine Books of Hadith [180] which are teachings and sayings from the Prophet Muhammad. Experimental results showed that ANN is better than the SVM as a classifier. The dictionary-lookup stemming method performed better with the ANN whereas the LS method performed better with the SVM classifier.

Al-Shammari's [163] compared LS [181], RE [182], and a proposed stemming method called "local stem". The local stem method selects the shortest form of a word among syntactically related words in a document. To evaluate the ATC system with the different stemming methods, Al-Shammari used a dataset that was constructed by merging the Saudi Newspapers (SNP) dataset [161] and the Saudi Press Agency (SPA) dataset [161]. Only 2966 documents are selected, and they

belong to three categories of “cultural”, “social”, and “general”. In the classification experiments, a 10-fold cross-validation was used with NB, SVM, and k-Nearest Neighbours (k-NN) ML algorithms. The experiment results showed that the local stemming method significantly improved text classification accuracy in comparison to other stemming methods. In addition, the higher performance came from the local stemming method with SVM classifier over the NB and k-NN classifiers which yielded an accuracy of 73.2%.

Other researchers compared the impact of the pre-processing component on the Arabic ATC system from other aspects such as the size of the document representation vectors, the classification time, as well as the accuracy of the ATC system. For example, Duwairi et al. [183] compared the original dataset, in which the words of all documents were used to build the BOW with a BOW based Root Extraction (BOW-RE), a BOW based Light Stemming (BOW-LS), and a Word Cluster Vector (WCV). To build the BOW-RE they used the tri-literal RE method proposed by Al-Serhan et al. [184]. A weight is assigned by the tri-literal RE method where the letters of each word are multiplied by the letter’s position in that word. For the BOW-LS, they applied Aljlayl and Frieder’s LS method [149]. To build the WCV, all synonymous words are grouped into a cluster labelled with the most frequently used word. The authors used in-house datasets collected from online resources that were prepared manually by filtering and classifying documents into three categories consisting of 5000 documents namely politics, sports, and economics. The k-NN classification algorithm is used to analyse the performance of the Arabic ATC system. The experiment results showed that BOW-RE had the shortest pre-processing time to build the representation model where the longest pre-processing time was took by WCV. In term of the time required to build the classification model, the BOW-RE again took the shortest time while the original BOW took the longest. In addition, experiment results showed that BOW-LS was the most accurate whereas the original BOW was the least accurate.

As can be seen in Table 6.8, a summary of the previous studies that mention earlier are listed where stemming method that used, representation model, classification algorithm/s, dataset and the experiment results are provided in the Table.

Table 6.8 Summarization of different studies comparing different stemming methods in ATC for Arabic text

Work	Stemming method	Representation model	Classifier	dataset	#of classes	# of training and testing docs	Performance
Harrag	LS [179] RE [177] dictionary-lookup [178]	BOW-LS BOW-RE BOW- dictionary lookup	ANN SVM	in-house dataset with 453	14	302 train 151 test	ANN yields 50% Macro-averages of F1 with BOW- dictionary lookup SVM yields 48% Macro-averages of F1 with BOW- LS
Al-Shammari [163]	LS [181] , RE [182] local stemming method		NB, SVM, and k -NN ML	(SNP) (SPA) [161]	3 2,966 doc's	10 fold cross-validation	SVM yield an accuracy of 73.2%. local stemming
Duwairi et al. [183]	RE (tri literal) [184]. LS [149]	BOW-RE BOW-LS	k -NN	in-house datasets with 15,000	3	9000 train 6000 test	k -NN yields 92% and 91% with BOW-LS for averages precision and recall, respectively

Other researchers applying classification algorithms to Arabic ATC such as El-Kourdi et al. [185] investigated the NB classification algorithm when it is applied to an Arabic ATC. An in-house dataset is created by collecting 1500 web pages covering five categories from the Al-Jazeera news website. The Arabic text was pre-processed by removing stop words and vowels. In addition, for each word, a root is extracted using the RE method as proposed by El-Kourdi et al. [186]. Two sets of experiments are run using cross-validation, one without the Features Selection (FS) technique and the other by using TFIDF as an FS technique and limiting the number of features to 50, 100, 500, 1000, and 2000 words which best represent all categories. The result showed that the NB classifier yields a higher accuracy of 68.78%, when all words are used, while it yields 62% by using only 2000 words.

The k -NN ML algorithm has been used by Kanaan et al. [187] to classify Arabic documents after pre-processing the text by removing the stop words and applying the LS method. To reduce the feature space the Information Gain (IG) FS technique is used. The authors used an in-house dataset by collecting 600 Arabic newspaper articles that fall into six categories from Al-Jazeera, Al-Safeer,

Fares.net, and Al-Dostor. Different sets of experiments are conducted by using different sizes of training and testing subsets. The experiment results show that the k-NN performance improves by increasing the size of the training subset, for example, when 360 documents are used as a training subset the performance improved by 20% in comparison to using 60 documents as the training subset.

The Decision Tree (DT) classifier has been used by Harrag et al. [188] to classify Arabic text after pre-processing the text by removing stop words and using a stemming algorithm on the remaining words. Then, the remaining words are weighted using the TFIDF. The authors proposed a hybrid approach for FS, which used embedded IG with DF as a thresholding. Two Arabic in-house datasets are used in the experiment. The first dataset consists of 373 scientific Arabic articles belonging to eight categories, where the second one is a set of 453 Islamic articles that fall into fourteen categories. To train the classifier, two-thirds of each dataset was used with the remaining portion used for testing. The experiment result shows the effectiveness of the proposed FS technique to improve the DT classifier. The ATC system performance is improved by 11% and 28% of average F1 measure for Islamic and scientific datasets respectively, compared to not using FS.

The performance of the SVM ML algorithm has been investigated by Mesleh [165] with seventeen different FS techniques which are GSS score (Gss) [189], GSS Square (GSSS) [190], NGL coefficient (NGL) [191], Odds Ratio (OR) [192], Mutual Information (MI), IG, Bi-Normal Separation (BNS) [193], DF, Power (Pwr) [193], Accuracy (Acc), Accuracy2 (Acc2) [193] and F1 measure [193]. An in-house dataset is collected from the online Arabic newspaper archives including Al-Jazeera, Al-Nahar, Al-Hayat, Al-Ahram, Al-Dostor and a few other specialized websites. The dataset contains 7842 documents that vary in length and fall into 10 categories. The results show that the Chi-square (χ^2) FS with SVM achieved better results than all other FS techniques which are for Arabic ATC.

As can be seen in Table 6.9, different studies in Arabic ATC system used different ML algorithms, but it is hard to compare them as each study used in-house datasets and different stemming methods.

Table 6.9 A summary of works that used different ML algorithms in Arabic ATC system

Work	Stemming method	Representation model	Classifier	Dataset	#of classes	# of training and testing docs	Performance
El-Kourdi et [185]	RE [186]	BOW-RE BOW-RE with FS	NB	in-house dataset with 1500	5	1500 train evaluation set with 10 docs	NB yield 68.78%, Accuracy with cross-validation for BOW-RE with 2000 words and it yields 62%, accuracy with evaluation set for BOW-RE with 2000 words.
Kanaan et al. [187]	Stemmed method used (not specified)	BOW-RE	<i>k</i> -NN	in-house dataset with 600	6	different sizes of training and testing (60, 120, 180, 240,360)	<i>k</i> -NN yield 62.7% and 79.3% with BOW for averages Macro precision and recall, respectively with 60% of dataset.
Harrag et al. [188]	LS [149, 177, 179, 182, 194, 195]	BOW-LS	DT	373 scientific Arabic articles 453 Islamic articles	8 14	66% training 33% testing	Improved the DT classifier by 11% and 28% average F1 measure for Islamic and scientific datasets respectively, compared to not using FS
Mesleh [165]	no stemming is applied	BOW 17 method of FS are used to select 10, 20, 50, 100, 200, 500, 1000, 2000, 4000 and 6000.	SVM NB <i>k</i> -NN Rocchio	in-house with 7842	10	4 - fold cross validation	SVM yields 91.410 % Macro-average F1 score with 1000 words used FS (χ^2)

Other researchers compared the accuracy of using different classification algorithms in Arabic ATC systems. For example, Mesleh [138] used the original data without using any stemming methods to build the BOW model; χ^2 was used as the FS technique to reduce the feature space. The dataset is collected from online Arabic newspaper archives and contains 1445 documents that vary in length and fall into nine categories. The results show that using an SVM in this context yields a better classification performance compared to *k*-NN and NB. It

yields a Macro-average F1 score of 88.11% when evaluated using an in-house Arabic text dataset.

The same dataset compiled by Mesleh [138] is used by Kanaan et al. [60] after using the LS method proposed by Aljlal and Frieder [149] and Larkey et al. [179] to build a BOW model for Arabic ATC. A comparison is performed between k-NN, Rocchio, and NB as classifiers with different weighting scheme such as Term Frequency (TF), TFIDF, and Weighted Inverse Document Frequency (WIDF). The results of this experiment show that the WIDF scheme yields the best performance when used in conjunction with k-NN, while TFIDF shows the best performance when used in conjunction with Rocchio. Among the three classifiers, the NB classifier is reported to be the best yielding a Macro- averaging an F1 score of 84.53%.

Al-Harbi et al. [161] compared the SVM and C5.0 (improved Decision Tree) classification algorithms in the Arabic ATC system. The original text is used without any stemming to build the BOW model and to reduce the feature space χ^2 is used as an FS technique. The C5.0 shows a better performance than the SVM. Where Al-Saleem [162] compared SVM and NB classification algorithms using the BOW model to classify an SNP dataset collected by Al-Harbi et al. [161] the experiment result showed that the SVM algorithm outperforms the NB algorithm.

Duwairi [196] compared the Distance Based (DB) classifier, k-NN, and NB to classify Arabic texts after it was pre-processed by removing stop words and punctuation marks, then Duwairi used the tri-literal RE stemming method [184]. Duwairi used an in-house dataset consisting of 1000 Arabic documents in ten categories. He concluded that the NB classifier achieved the best classification accuracy in comparison with the other two classifiers.

Others such as Al-Saleem [197] compared the traditional ML algorithms and Classification Based Association (CBA) to classify Arabic documents. Al-Saleem used an Arabic SNP text collection compiled by Al-Harbi et al. [161] a dataset consisting of 5121 Arabic documents in seven categories, to apply SVM, NB, and CBA as classification algorithms. The results show that CBA outperformed NB and SVM, and its average F1 score was 0.804.

The in-house dataset used by Hmeidi et al. [198] contains 2206 articles for training and 29 articles for testing which fall into two categories. These articles

were collected from the Al-Rai and Al-Dostor newspapers which are the most common newspapers in Jordan. The original texts were used without any stemming to build a BOW model to study the performance of SVM and k-NN ML algorithms on Arabic ATC. To reduce the feature space, χ^2 was used as an FS technique to select 50-500 words. Their results show that both classifiers worked well in the Arabic ATC system. The SVM classifier in this context yields better classification performance in terms of prediction time and F1-measure compared to the k-NN classifier. When 450 words are selected, the SVM yields an average F1 score of 0.97.

Hadi et al. [199] compared the performance of SVM and NB in Arabic ATC systems. The experiment used an Islamic dataset collected by Al-Harbi et al. [161] consisting of 2244 Arabic documents of different lengths that fall into five categories. To build the BOW model, the pre-processing steps from [185, 200-202] are followed. To evaluate the performance of the classifier in the Arabic ATC system, a 10-fold cross-validation is applied. The results show that SVM was superior to NB as the classification performance of SVM yielded an average F1 score of 0.95 while NB yielded an average F1 score of 0.884. A summary of these studies can be seen in Table 6.10. The same datasets are used in different experimental setups to evaluate ATC of Arabic text.

Table 6.10 A summary of different works that used same datasets with different ML algorithms in Arabic ATC system

Work	Stemming method	Representation model	Classifier	dataset	#of classes	# of training and testing docs	Performance
Mesleh [138]	no stemming is applied	BOW FS (χ^2) with 162 words per class	SVM NB k -NN	Arabic 1445	9	66% train 33% test	SVM yield 88.11% Macro-average F1 score
Kanaan et al. [60]	LS [149] [179]	BOW-LS	k -NN Rocchio NB	Arabic 1445 [138]	9	10-fold cross-validation	NB yielding a Macro- averaging F1 score of 84.53%.
Al-Harbi et al. [161]	no stemming is applied	BOW FS (χ^2) with 30 words per class	SVM C5.0 (DT)	SPA SNP WEB Writer Forms Islamic poems	6 7 6 10 7 5 6	70% train 30% test	C5.0 yielding averaging accuracy of 78.42% for all dataset C5.0 yielding averaging accuracy of 79.49% for SNP dataset where SVM yields 72.73%
Duwairi [196]	RE (tri-literal) [184]	BOW-RE	DB k -NN NB	in-house dataset with 1000	10	50% train 50% test	NB yields 80% and 79.3% with BOW for averages Macro precision and recall, respectively, with 60% of dataset.
Al-Saleem [197]	Not found paper		SVM NB CBA	SNP[161] 5121	7		CBA yielding averaging F1 score of 80.4%.
Hmeidi et al. [198]	no stemming is applied	BOW with FS (χ^2) used to select 50-500 Words	SVM k -NN	in-house dataset with 2235	2	2206 train 29 test	SVM yielding averaging F1 score of 97% with 450 words
Hadi et al. [199] (2010)	no stemming is applied	BOW	SVM NB	Islamic dataset [161] 2,243	5	10-fold cross-validation	SVM yielding averaging F1 score of 95%

In terms of text representation models, most of the work in the Arabic ATC system have used the BOW model where others used various features to represent Arabic text in the ATC system. For example, Khreisat [71] used an N-gram frequency statistical technique to compare two similarity measures, Manhattan distance, and Dice's coefficient, in the Arabic ATC system. The author used in-

house datasets collected from four online Jordanian Arabic newspaper archives. The tri-gram was used to represent each document after pre-processing by removing punctuation marks, diacritics, non-letters, and stop words. The Khoja RE method [182] is applied on remaining words. To classify Arabic text, the similarity measures are used with 40% of the dataset as a training subset and the rest for the testing subset. The results show that the best accuracy was obtained using Dice's coefficient in conjunction with the tri-gram frequency method.

Others used statistical phrases to represent Arabic text in ATC such as Al-Shalabi and Obeidat [203] where the authors compared BOW and phrases text representation models in Arabic ATC. The k-NN ML algorithm is used to classify documents from the dataset collected by Mesleh [138]. All Arabic documents are pre-processed by removing stop words, non-letters, and punctuation marks, then two independent weighted representations are built, the BOW and phrases (two-words). To train the classifier, 60% of the dataset is used where the rest is used for testing. The results show that using phrases (two-words) as a text representation model when used in conjunction with the k-NN classifier outperforms the BOW model by yielding an average accuracy score of 73.57% where the BOW model yields only 66.88%.

The Arabic WordNet is used by Elberrichi and Abidi [164] to identify concepts appearing within the documents. A comparison between different text representation models such as BOW, N-grams, and combined models constructed by using RTC strategy was conducted using an Arabic text dataset collected by Mesleh [138]. The combined model, used in conjunction with χ^2 as FS technique and a k-NN classifier, was reported to achieve higher performance results compared to other representations. A summary of these studies can be seen in Table 6.11, where different types of features are used to represent Arabic text in ATC.

Table 6.11 Different representation models used in Arabic ATC

Work	Stemming method	Representation model	Classifier	dataset	#of classes	# of training and testing docs	Performance
Khreisa t [71]	RE [182]	N-gram (tri-gram)	DB	in-house datasets	4	40% train 60% test	DB with Dice measures yielding 0.83% and 0.88% with 3-gram for averages recall and precision, respectively
Al-Shalabi and Obeidat [203] 2008	no stemming is applied	BOW Statistical phrases (two-words).	<i>k</i> -NN	Arabic 1445 [138]	9	60% train 40% test	<i>k</i> -NN with phrases yielding average accuracy of 73.57%
Elberrihi and Abidi [164] 2012	no stemming is applied	BOW N-grams Combined model using RTC	<i>k</i> -NN	Arabic 1445 [138]	9	66% train 34% test	<i>k</i> -NN with RTC yielding F1 score of 0.74

6.5 Arabic ATC System Design

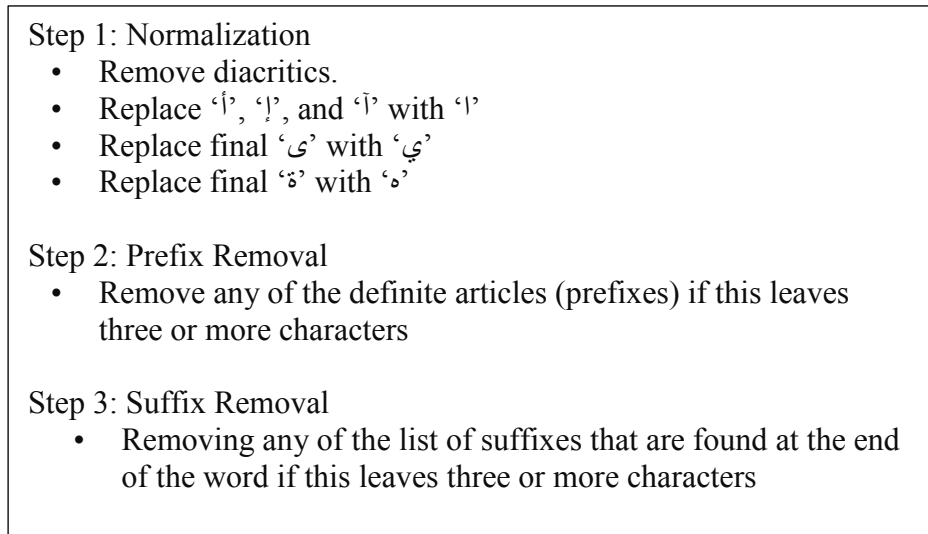
To evaluate any language in text mining, it must be part of a particular target application such as an ATC system. In this section, the different representation models are evaluated with Arabic text in an ATC system in terms of their efficiency. The higher classification quality outcome of an ATC system indicates a more effective representation model. Three Arabic datasets (see Section 6.3) were used to evaluate the different components of the ATC system with Arabic text.

First, all documents in the dataset are converted to lower case and for each dataset, the correct encoding is chosen to read the documents. Table 6.12 shows each dataset and the correct encoding.

Table 6.12 Different datasets and their encoding

Dataset	Encoding
Arabic 1455	UTF8
SNP	Windows-1256
Al-Khaleej	Windows-1256

Each document in the dataset is then tokenized into words where white space is used as a word separator. To compare different methods in pre-processing three different BOWs are constructed: BOW-RE, BOW-LS, and original BOW. In the original BOW, each word is kept the same whereas with the BOW-RE each word is stemmed by extracting the root with the Khoja stemmer [182]. For the BOW-LS, each word is stemmed following the method outlined by Aljlal and Frieder [149] as shown in Figure 6.1 [60].

**Figure 6.1 Algorithm for LS method**

All stop words are then removed using the 164 stop word list [204]. In addition, rare words which appear less than a certain number of times in the whole dataset are removed. In our experiments, we removed all the words that appeared less than four times in the dataset. The reason behind selecting four as a threshold for our system is based on previous works that reported an improvement on the classification accuracy and reduction of the high dimensions of feature space in the model [60, 138]. The remaining words are used to represent documents using

the VSM, and each word is weighted using the TFIDF weighting scheme (see Section 2.2.2).

For the BOC, both Wikipedia and WordNet are used as a KB and BOC-based Wikipedia and BOC-based WordNet are compared to find the best KB for this experiment. To use Arabic WordNet for identified concepts in the text, an Arabic WordNet (AWN) Browser open-source toolkit [205] (an automatically-updated, locally-stored database) is used. For each word in a text, the AWN Browser searches the database and returns an ordered list of synonyms and the first ranked synonym will be used as a concept for the word and used to represent text in the BOC-based WordNet. To build the BOC-based Wikipedia, the Arabic Wikipedia XML dump files (dated 2014-01-12), which comprise 273,709 articles [206], are used in this thesis. To process the dump files, an open source toolkit known as Wikipedia-Miner [139] is used to create a database, which contains a summarized version of Arabic Wikipedia's content and structure. To achieve that, we used a Hadoop cluster with two 3.33GHz processors and 16GB of RAM. Wikipedia-Miner processed the dump files in 12 hours. The toolkit assigns a unique identification, title, and type (article, category, or redirect) to each Arabic Wikipedia page.

The first step is to process and clean a document. Then, all possible Wikipedia concepts in the document will be identified using the topic detection functionality in Wikipedia-Miner. The majority of detected concepts are ambiguous as they may refer to multiple meanings. To disambiguate these concepts, a supervised disambiguation algorithm by Milne and Witten [115], which uses bagging [49] of C4.5 decision trees [45], is trained with 100 randomly sampled Wikipedia articles. The disambiguation algorithm computes a probability estimate for each potential concept based on its relevance to other detected concepts in the document. The probability determines if a concept is relevant to the context or not and the ambiguity is resolved. A set of candidate concepts are selected which are used to represent documents in the BOC model by VSM, and each concept is weighted using the TFIDF weighting scheme.

Another text representation model is built by combining the BOW and the BOC; the combined model enriches the BOW model with concepts from the Arabic

Wikipedia using different strategies. These strategies are AC, RTC, ACC, AUC, and CTD (see Section 3.6 and 4.2).

Most work with Arabic ATC uses one or two classification algorithms on single datasets where different classification algorithms have different behaviours that might impact the performance of the system with different representation models. For that, Support Vector Machines (SVM), Naive Bayes (NB), Decision Trees (DT), and Random Forest (RF) classifiers are used to compare their performance with different representation models. A 10-fold cross-validation is performed for all datasets.

6.6 Experimental Result & Evaluation

In this section, the experimental results of three main areas are presented and analysed:

- The impact of the pre-processing component on the BOW.
- The impact of the characteristics of the knowledge source used to build the BOC.
- Comparison of different strategies used to build the combined representation models.

First, we conduct a comparison between the original text and the different methods used in the pre-processing component to stem a word, namely LS and RE. Second, the two KBs used to build the BOC, WordNet and Wikipedia, are investigated in the Arabic ATC. Third, the five combined representation models for ATC in Arabic are evaluated.

6.6.1 Stemming Methods

To build the BOW for the Arabic ATC system, the Arabic text is first tokenized, and the stop words are removed. In the case of building an original BOW, the remaining words are used to build the dictionary for the BOW. If the remaining words are matched to existing words in the dictionary, the frequency for existing words are increased. Otherwise unmatched remaining words need to be added as new features in the dictionary. In the other case, when stemmed BOW is selected,

before building the BOW dictionary each word is stemmed. Two stemming methods, LS and RE are compared. Based on the stemming method used, two BOW models, BOW-LS and BOW-RE are built.

As a result of the stemming process, for each dataset, three various BOW model representations, BOW-LS, BOW-RE and the original BOW, are built. Each one of these BOW model representations will be used with four classification algorithms. Furthermore, these BOW models have a different number of features which leads to different times that the classifier takes to classify a document and produced a different accuracy performance for ATC. Figure 6.2 shows the three datasets, their various BOW models, and the accuracy they achieved using SVM, NB, DT, and RF classifiers. As can be seen in Figure 6.2 the SVM classifier achieved the highest accuracy followed by NB, DT, and RF.

The SVM classifier achieves a higher accuracy when it is used with the original BOW for both SNP and the Al-Khaleej datasets which are considered to be quite large. The small dataset, Arabic 1455, achieved the highest result with BOW-LS and the SVM classifier. The SVM classifier shows its ability to deal successfully with high-dimensional data [31].

The NB classifier achieved a higher accuracy with the BOW-RE for both Arabic 1455 and Al-Khaleej datasets while the SNP dataset achieved the higher accuracy with BOW-LS representation. The RF classifier achieved the best accuracy with original BOW for all datasets. The DT classifier also achieved the highest accuracy with original BOW for both Arabic 1455 and SNP datasets while Al-Khaleej achieved the highest result with BOW-LS for the same classifier.

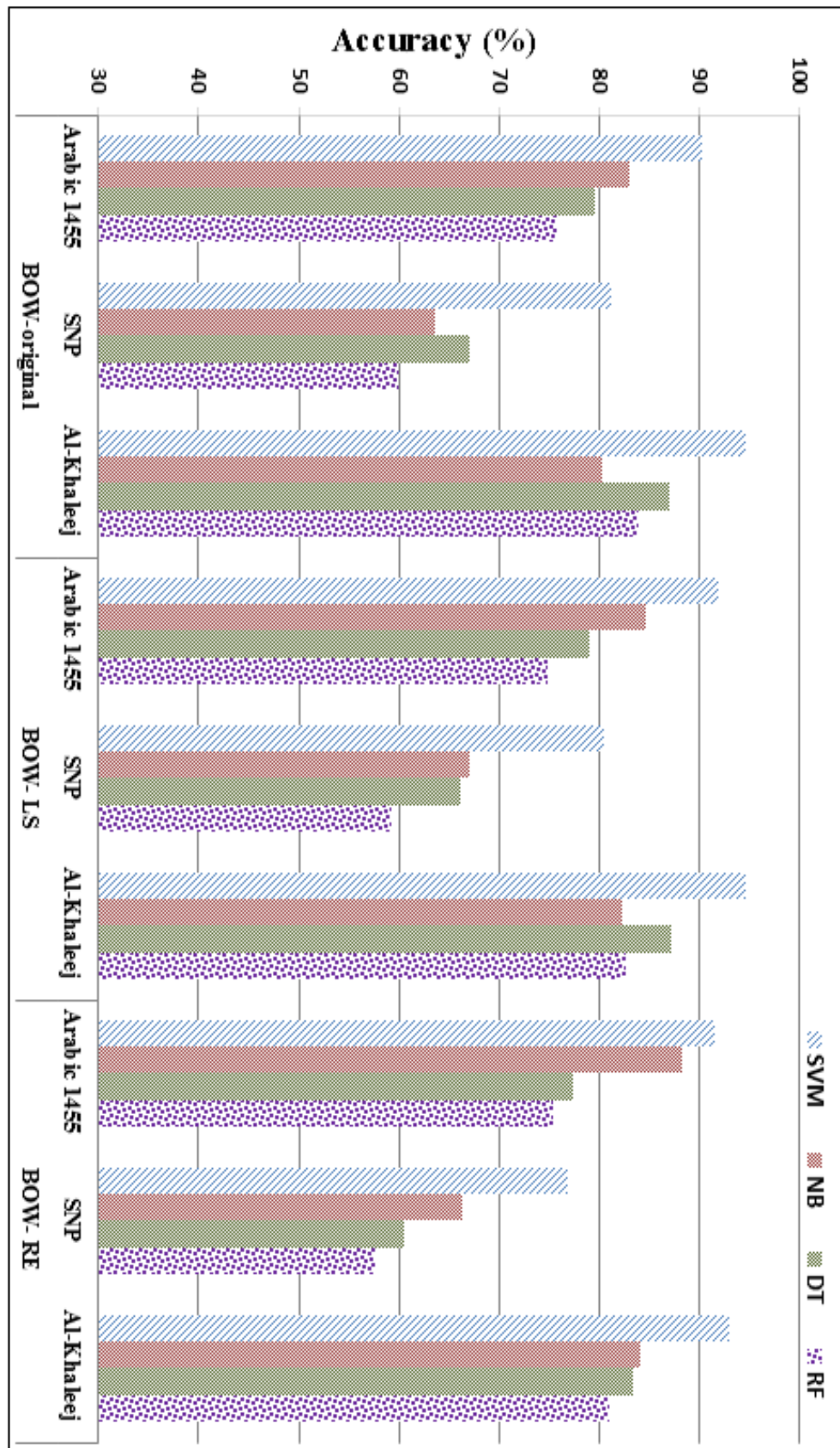


Figure 6.2 Classification accuracy of different BOW models with different classifiers in Arabic ATC

In terms of the size of the representation model Figure 6.3 shows the number of features in each representation model, original BOW, BOW-LS, and BOW-RE. In all datasets, the smallest representation model is provided by the BOW-RE representation model with only 2553 features for the Arabic 1455 dataset, 2488 features for Al-Khaleej, and 2586 features for SNP.

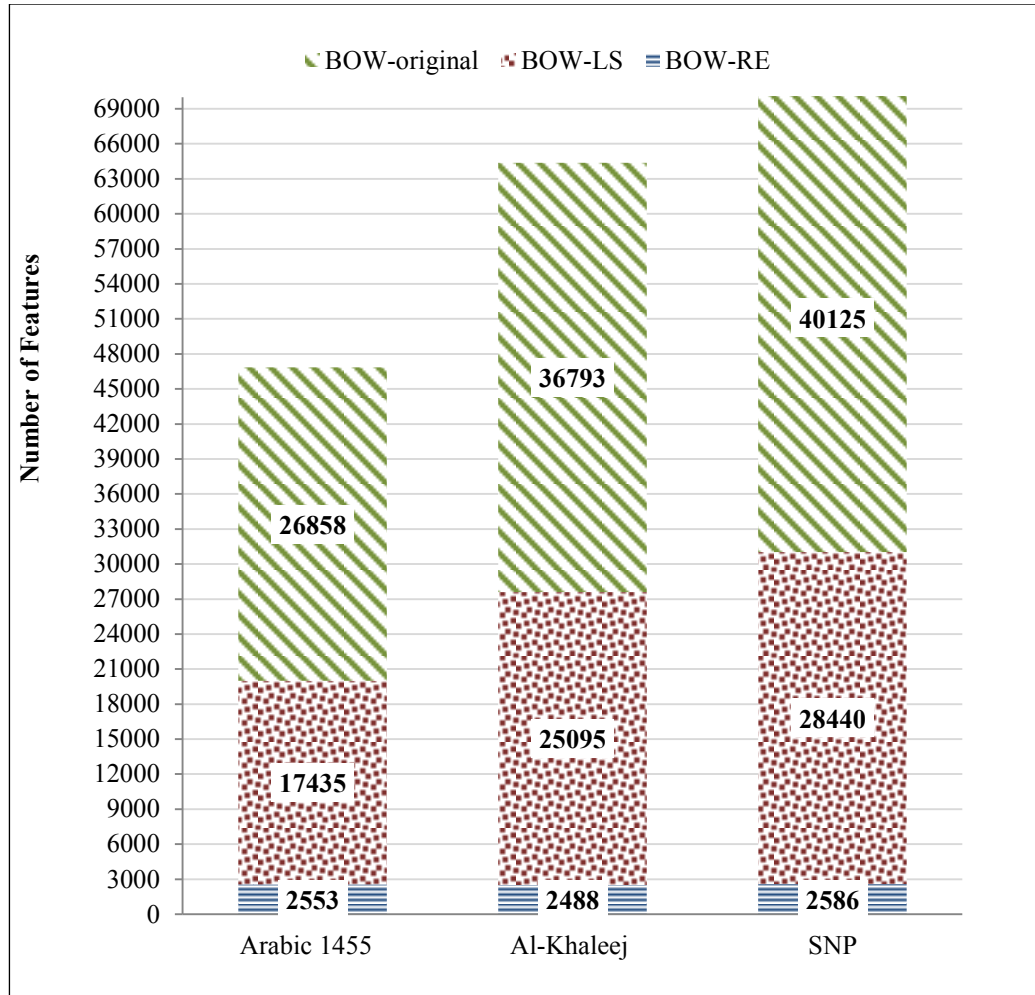


Figure 6.3 Number of features in representation models in different dataset

By using the RE stemming method in the pre-processing component with Arabic text, the size of the original text is reduced by 90% in the Arabic 1455 dataset, 93% for the Al-Khaleej dataset, and 94% for the SNP dataset. On the other hand, using the LS method only reduced the size of the original by 35% in the Arabic 1455 dataset, 32% for the Al-Khaleej dataset, and 29% for the SNP dataset.

The time that the classifier takes to train and classify documents is affected by a number of issues such as the number of features in the representation models, the size of the dataset, and the type of the classifier. The larger the dataset, the more

features it produces and the longer it takes to train a classifier and classify a document. In our experiment, we used a machine with two 3.33GHz processors and 16 GB of RAM.

Figure 6.4 shows the time required by SVM, NB, and RF classifiers whereas Figure 6.5 shows the time required by the DT classifier with different BOW representation models in all datasets.

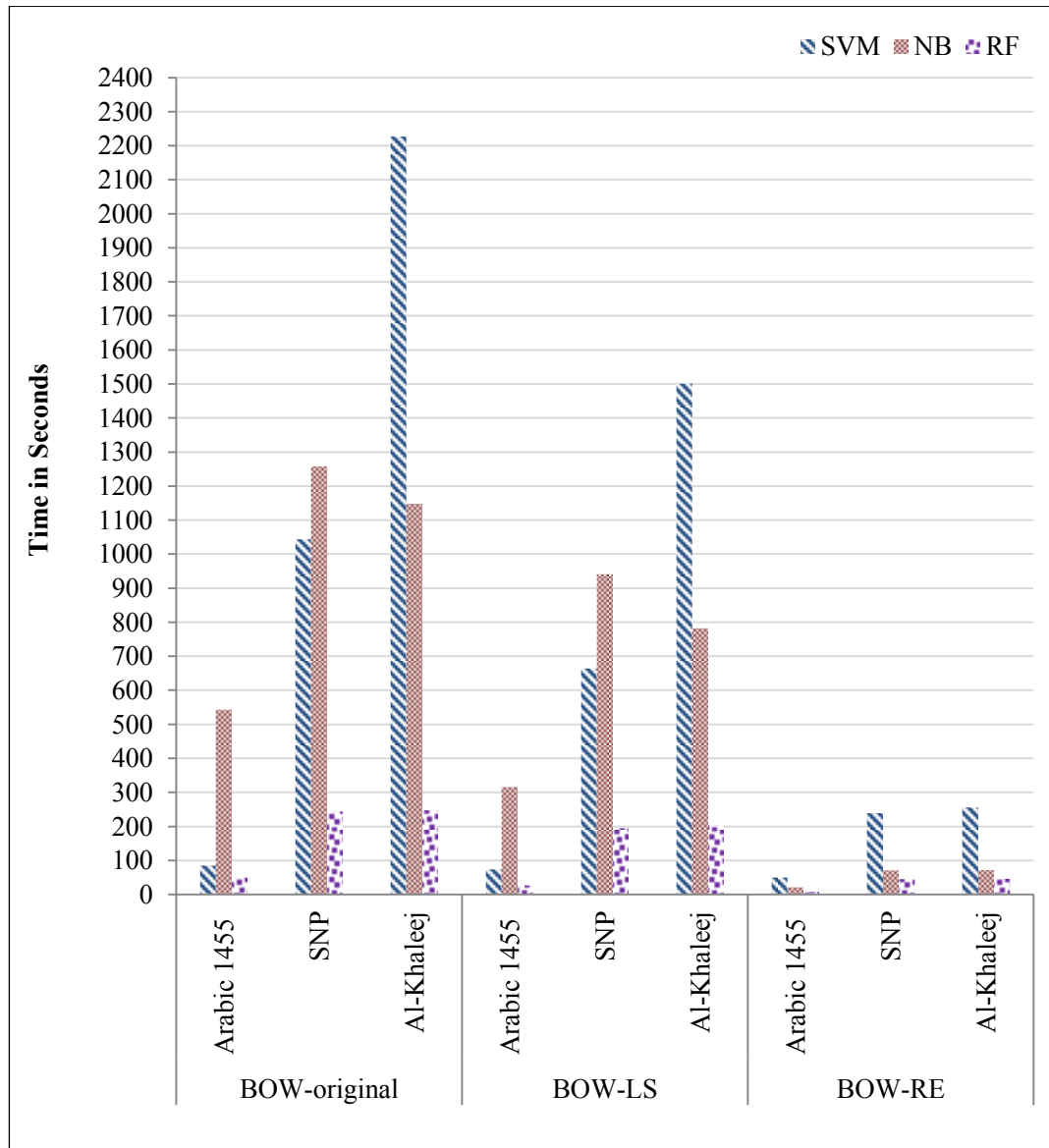


Figure 6.4 Time required by different representation models with SVM, NB, and RF classifiers

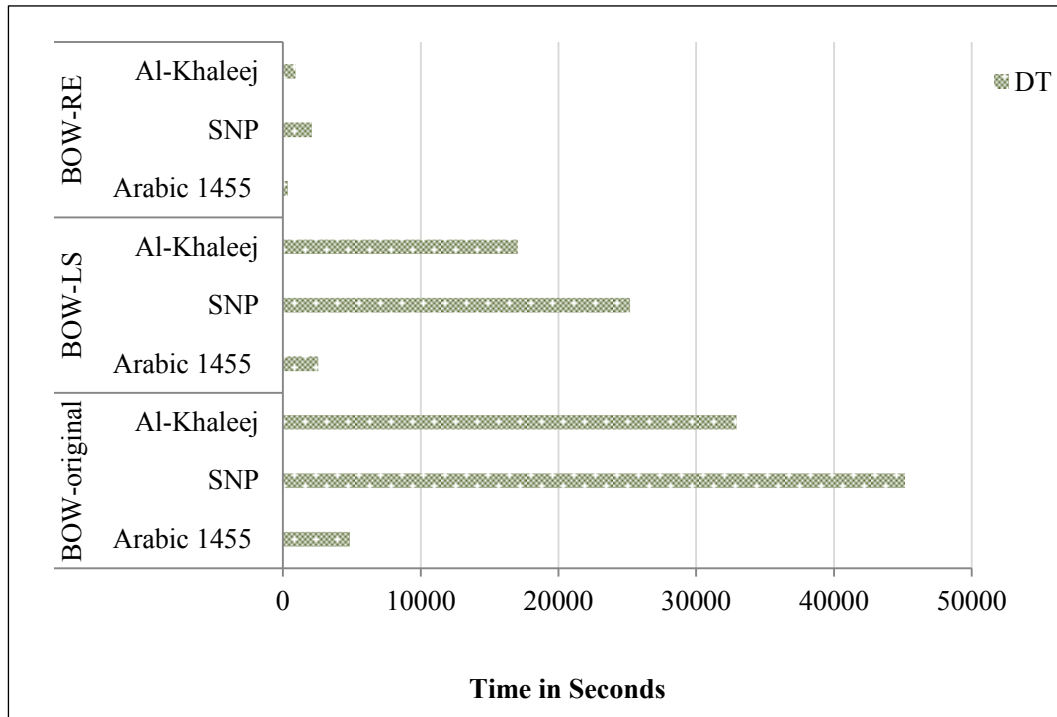


Figure 6.5 Time required by DT classifier with different BOW representation models.

From both Figure 6.4 and 6.5, it is found that the BOW-RE representation model takes the least time in all datasets with all classifiers. As expected, the smallest dataset, Arabic 1455, took the least time in all BOW representation models with all classifiers. The DT classifier took the longest time to train all different BOW representation models, while RF took the least time.

In order to find which stemming method works best with any of the classification algorithms we have considered, the performance of the three datasets is averaged as can be seen in Figure 6.6.

From Figure 6.6 it can be concluded that the SVM classifier achieved a higher average accuracy with BOW-LS while the NB classifier achieved a higher result with the BOW-RE classifier. Both DT and RF achieved higher accuracy with the original BOW.

In terms of accuracy, the SVM classifier performance with BOW-LS improved by 0.32 more than the original BOW, and improved by 1.83% more than the BOW-RE. On the other hand, the NB classifier performance improved with BOW-RE by 1.58% than the BOW-LS and the NB classifier performance improved by 3.96% than the original BOW.

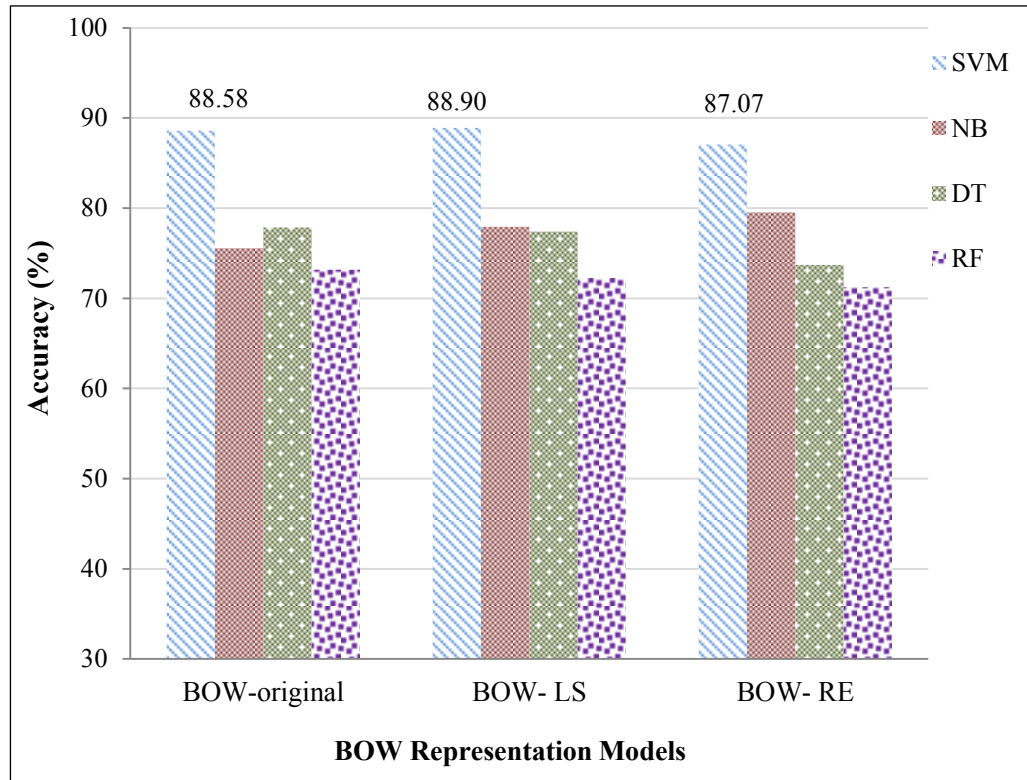


Figure 6.6 Average accuracy of different BOW representation models with different classifiers

As can be seen in Figure 6.7, the least pre-processing time is required by the RE followed by the LS method while the simplest way, the original BOW, took the longest time.

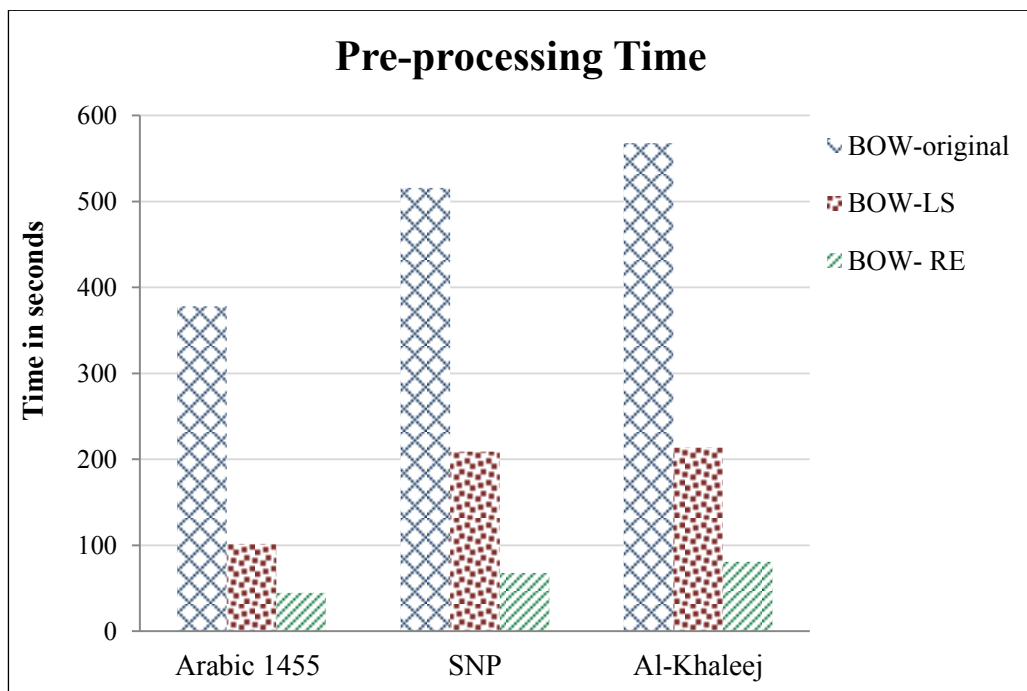


Figure 6.7 Pre-processing time with different BOW representation models

The reason behind this is that RE steps are faster, and it reduces the original BOW representation by 90%. On the other hand, even though the original BOW model does not involve any morphological processing, it needs more time than both BOW-RE and BOW-LS because of the high dimensionality of the original BOW model. The original BOW model takes a long time to search a huge number of features (high number of dimensions) to match words from the dictionary to represent text.

6.6.2 Knowledge Bases

To build a BOC model, a KB such as WordNet or Wikipedia is needed to provide concepts to represent text. In this section, a comparison between WordNet and Wikipedia is applied to the Arabic ATC system with three datasets. To the best of our knowledge, Wikipedia as a KB has not been used to build a BOC for Arabic ATC. In addition, using WordNet concepts alone to represent Arabic text has not been used for Arabic ATC. Figure 6.8 shows the performance of BOC for an Arabic ATC system with all datasets.

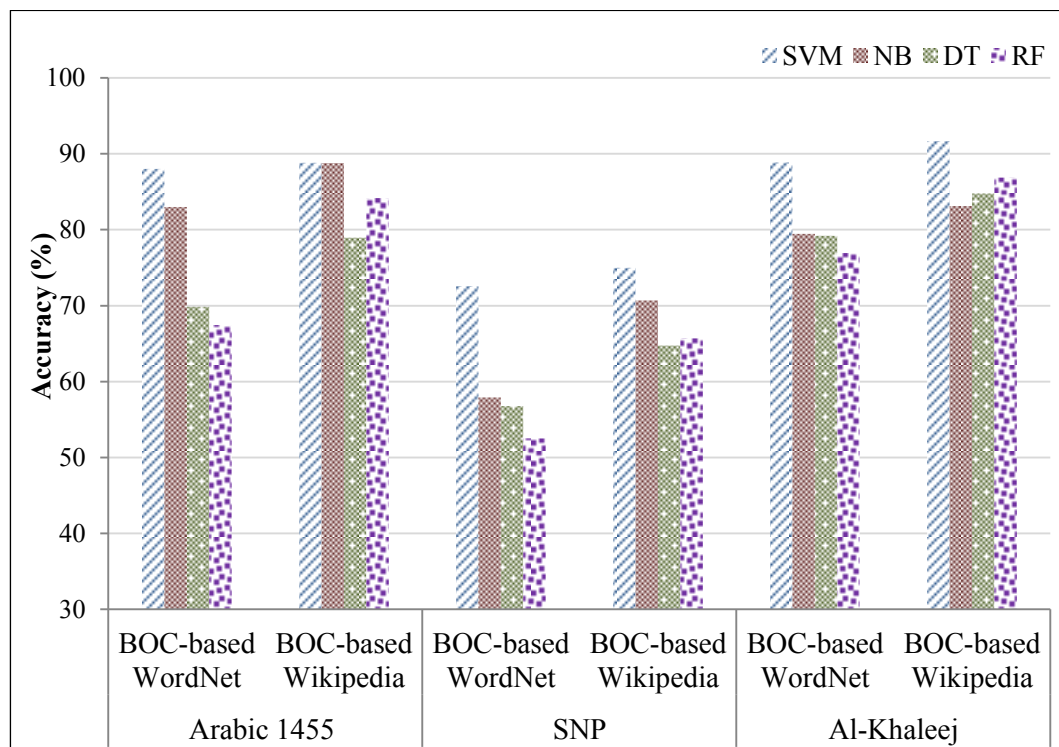


Figure 6.8 Classification Accuracy of BOC-based Wikipedia and BOC-based WordNet models with different classifiers in Arabic ATC

From Figure 6.8 it can be concluded that it is better to use Wikipedia to provide concepts to build a BOC for an Arabic ATC system than WordNet. All classifiers in all the datasets achieved a higher accuracy with Wikipedia as a KB to build a BOC than WordNet. One of the reasons behind this is that WordNet provides a BOC model with fewer features for representing Arabic text, as can be seen from Figure 6.9.

Arabic WordNet has only 9228 concepts where Arabic Wikipedia has 273,709 concepts defined. This means that WordNet has less concepts that can be used to represent Arabic text in ATC. Another reason is the ambiguity of the text as all the datasets are MSA texts which do not have any diacritics. Arabic WordNet returned a ranked list of possible concepts for a word in the text, and the first ranked is based on the most common uses where Wikipedia selects concepts based on surrounding text.

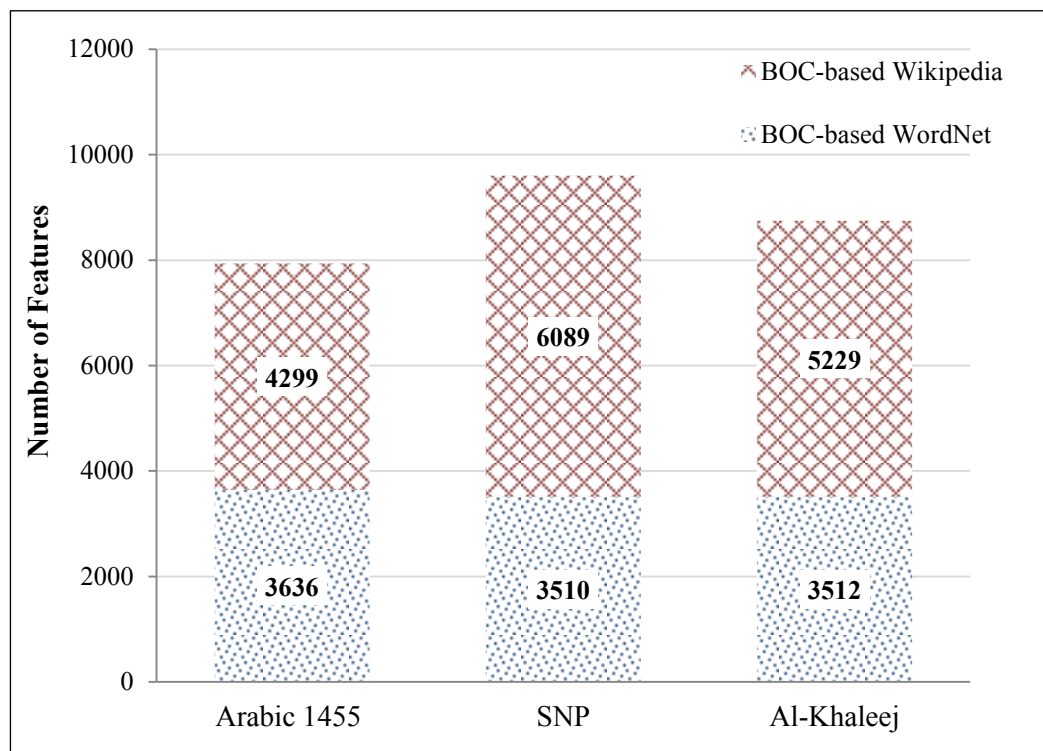


Figure 6.9 Number of features of BOC-based Wikipedia and BOC-based WordNet models in different datasets

Figure 6.10 shows the time required by SVM, NB, RF, and DT classifiers with both BOC-based WordNet and BOC-based Wikipedia models in all datasets. The longest time was taken by the DT classifier followed by SVM, then NB, and the quickest classifier is RF. Using Wikipedia to build a BOC model provides more

features than WordNet by 15%, 42%, and 33% for Arabic 1455, SNP, and the Al-Khaleej datasets respectively. That leads to a long classification time especially for the DT classification algorithm where for each split node to build the classification tree it searches all features for the best one to split the data.

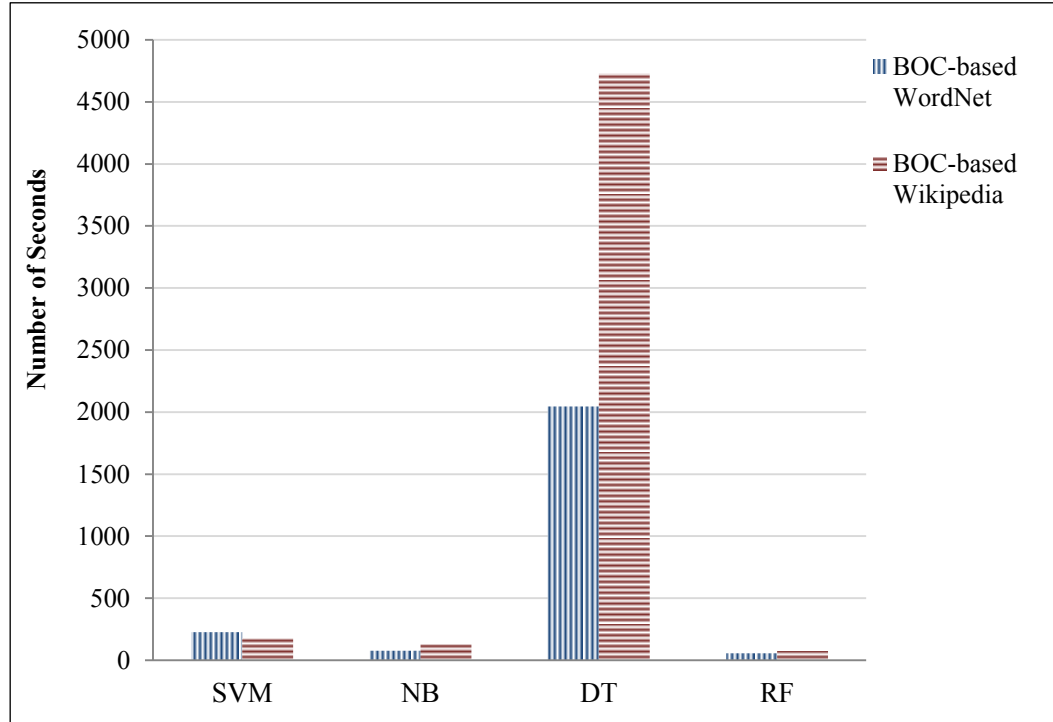


Figure 6.10 Time required by BOC-based Wikipedia and BOC-based WordNet models with different classifiers

6.6.3 Evaluation of Combined Representation Models

In this section, the combined representation models are evaluated in the Arabic ATC system with three datasets. To build the combined models, both features from BOW-LS and BOC-based Wikipedia are used to provide combined models by using different strategies to represent Arabic text with these features. In this section, the strategies AC, RTC, ACC, AUC, and CTD (see Section 3.6 and 4.2) are used.

As can be seen from Figure 6.11, the SVM classifier achieved higher accuracy in all combined models compared to other classifiers. The NB classifier comes next with 5.6% less accuracy than SVM, followed by DT, which is less than SVM and NB by 10.4% and 4.7% respectively. The RF classifier achieved the lowest

accuracy in comparison to SVM, NB, and DT by 13.36%, 7.75%, and 2.9% respectively.

In terms of combined models, the average highest accuracy was achieved by the combined model using the ACC strategy, followed by the AC, and the AUC. AC and AUC strategies were less accurate than ACC by only 0.25 and 0.46 respectively.

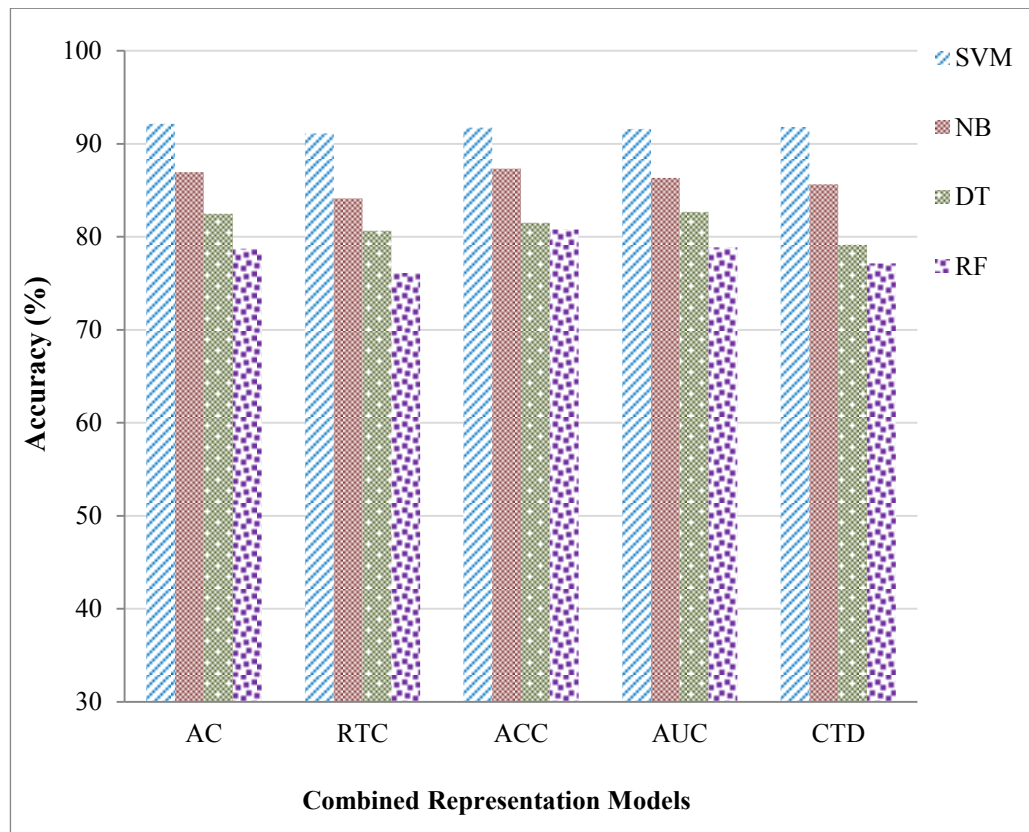


Figure 6.11 Classification accuracy of combined models with different classifiers in the Arabic 1445 dataset

The combined model using the CTD strategy has fewer features in comparison to other combined models which are using other strategies and has achieved an average accuracy of 83.45%, which is less than the highest average accuracy by only 1.9%. The reason behind the minor difference in accuracy of the combined representation models is because of the small number of documents in the dataset and all combined representation models used to represent Arabic text for TC show the same level of efficiency.

Figure 6.12 shows that both ACC and CTD strategies achieved the highest accuracy compared to other strategies when is used to build combined model

representation in the SNP dataset. The CTD strategy has fewer features in comparison to the ACC strategy. The combined model using CTD strategy only has 25% of the ACC combined model size. The fewer features the representation model has, the less time it takes to train a classifier and to classify documents. As a result, the CTD combined model is faster than the ACC combined model by 32%. By using NB as a classifier with the CTD combined representation model, this achieved the highest accuracy compared to other representation models.

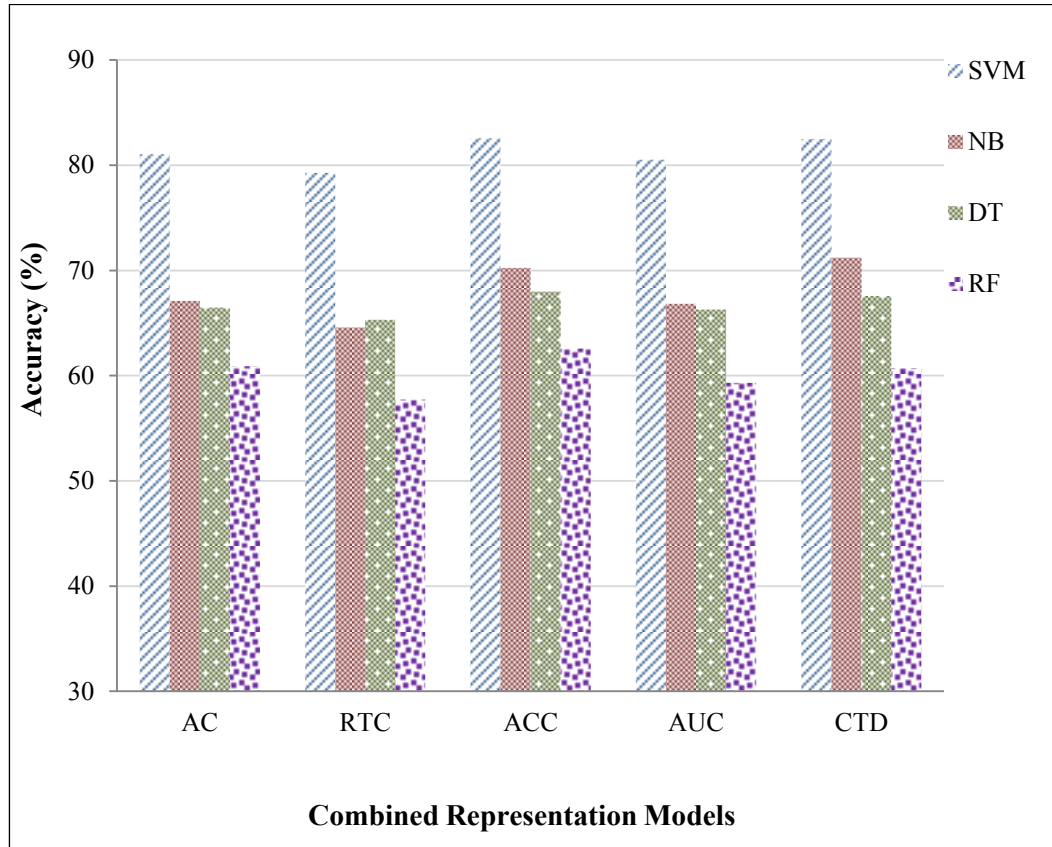


Figure 6.12 Classification accuracy of combined models with different classifiers in the SNP dataset

In addition, the SVM classifier achieved higher accuracy in all combined models compared to other classifiers. Both NB and DT classifiers come closely next with a small difference between them in term of classification accuracy where in the previous datasets the NB classifier outperformed the DT with a noticeable difference as can be seen in Figure 6.12.

It can be seen from Figure 6.13 that the SVM classifier achieved higher accuracy in all combined models compared to other classifiers. The DT classifier comes next with 6.7% less accuracy than the SVM, followed by NB, which is less

accurate than the SVM and DT by 11.3% and 4.6% respectively. In terms of classification accuracy, the AUC combined model achieved higher accuracy with the SVM classifier in comparison with others. The AC combined model comes next with 0.20 less accuracy than AUC, followed by RTC, which is less accurate than AUC and AC by 0.26 and 0.06 respectively. AC, ACC, and AUC with DT classifier come closely next with a small difference between them in terms of classification accuracy. The AC combined model achieved a higher accuracy with the RF classifier, followed by AUC, and RTC, which is less than AC by 0.33 and 0.50 respectively.

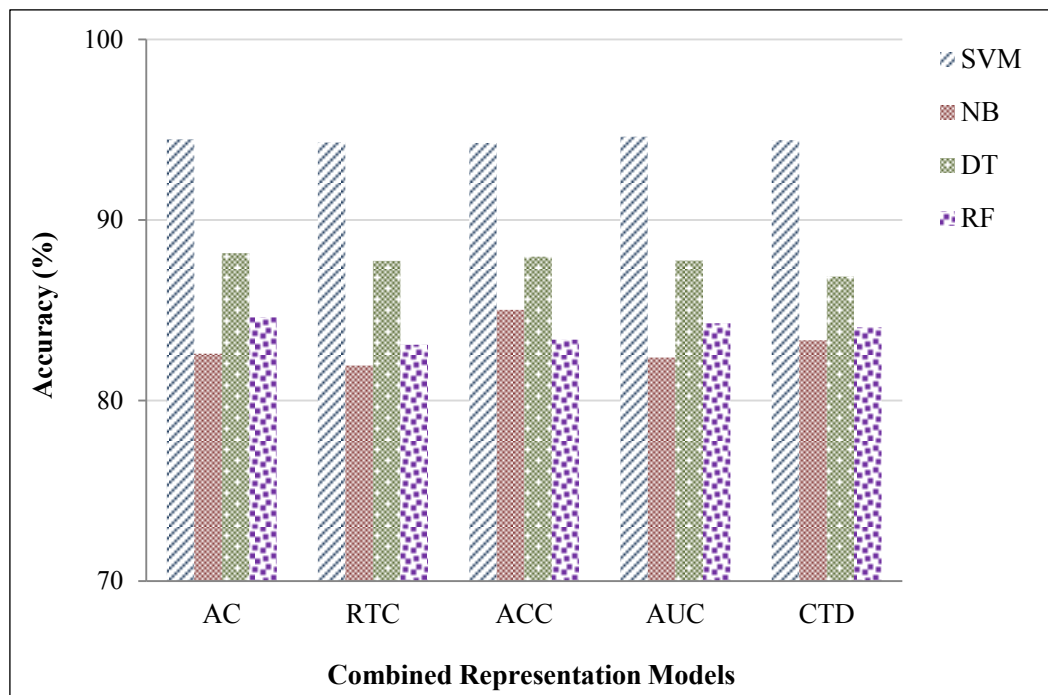


Figure 6.13 Classification accuracy of combined models with different classifiers in Al-Khaleej dataset

In terms of the number of features, it can be seen from Figure 6.14; the largest representation model is constructed using the ACC method in all datasets where the smallest is RTC followed by CTD. The SNP dataset has the largest number of features in all the combined representation models even though it has less number of documents than the Al-Khaleej dataset (see Section 6.3.1 and 6.3.3). The reason for this is that the SNP has more categories than Al-Khaleej, which leads to more distinguished features which require a longer time to classify documents than in other datasets.

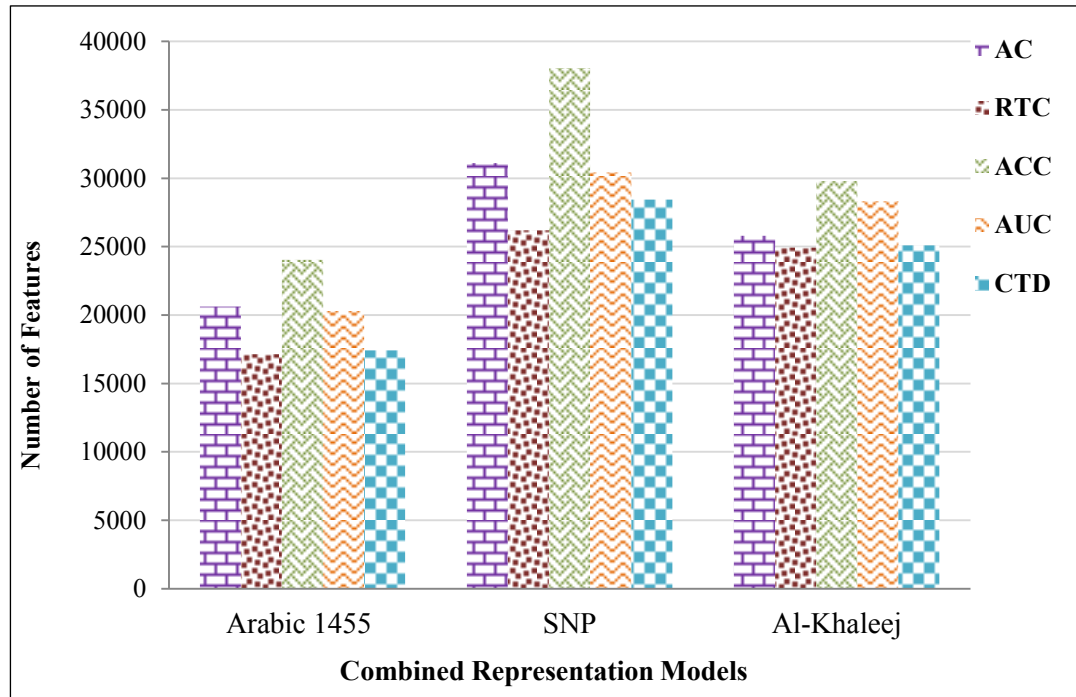


Figure 6.14 Number of features in the combined models with different datasets

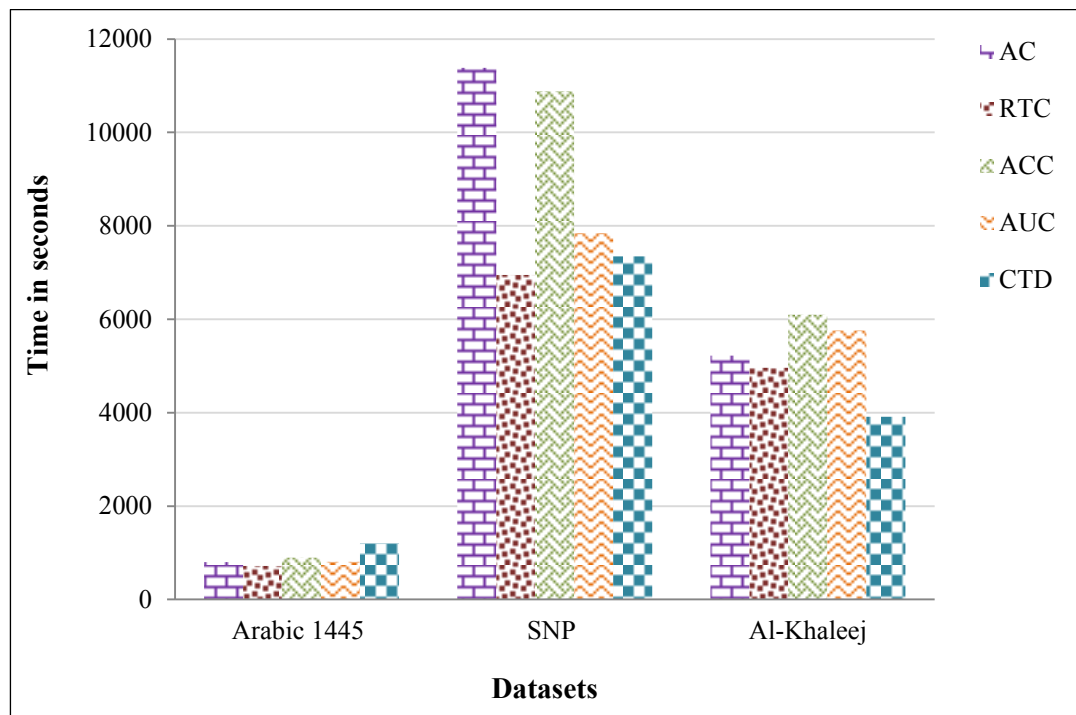


Figure 6.15 The classification time of different combined model in the three datasets

As can be seen from Figure 6.15, all combined models in the SNP dataset take the longest classification time in comparison to other datasets. The combined model using ACC strategy in both SNP and Al-Khaleej datasets take a longer classification time compared to others.

6.7 Discussion of Results

The aim of our experiments is to show how text representation models affect the performance of an Arabic ATC system; concepts and words as features are used in the experiments to achieve this. These features are used individually when creating the BOW (see Section 3.2) and BOC (see Section 3.5) by using a single type of feature or combining these features using different strategies such as AC, RTC, ACC, AUC, and CTD (see Section 3.6 and 4.2). In addition, two KBs, WordNet and Wikipedia, are used to provide concepts to represent text in BOC. Furthermore, using different methods for the stemming step in the pre-processing stage in ATC leads to the creation of various BOW models which are LS and RE, besides using original text without stemming.

By counting all these representation models, we have ten, which are BOW based, Original BOW, BOW-LS, BOW-RE, BOC-based WordNet, BOC-based Wikipedia, AC, RTC, ACC, AUC, and CTD. The performance of the Arabic ATC system with these representation models can be compared by accuracy, the size of representation models, and the speed of the learning process for classifiers.

In the previous section (see Section 6.6) the result of the classification accuracy is analysed for each dataset separately with different text representation and classification algorithms whereas in this section the classification accuracy results of three datasets are averaged to discuss the results.

To compare the stemming method in the pre-processing stage, we averaged the performance of the different representation models of the three datasets with different classifiers, as can be seen in Figure 6.16. The BOW-LS achieved the best accuracy when compared to using the original text or RE stemming method to build a BOW model.

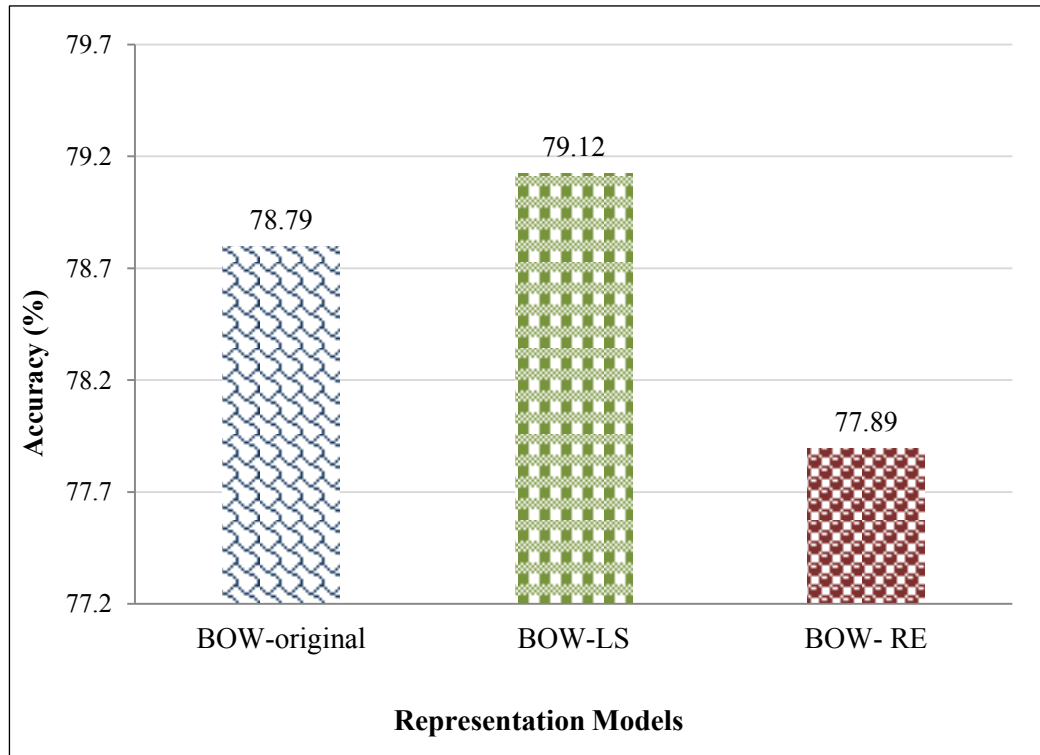


Figure 6.16 Average accuracy of various BOW models in all datasets with different classifiers

The BOW-RE provides fewer features in the representation model which leads to a faster learning time to build classifiers (see Section 6.6.1, Figures 6.4 and 6.5) but BOW-LS is considered the best representation model in terms of classification accuracy which is in line with other researchers findings such as [176, 183, 207].

The BOW-LS is found to be more accurate with large datasets than BOW-RE. The reason for this is that the RE stemming method is harsher on words in comparison to LS methods. For example, two words with independent meaning are stemmed to the same root even though they have a different meaning which leads to misclassified text. For example, both words (العالمية) “global” and (العلمية) “scientific” share the same root, (علم) “science”.

From Figure 6.17 it can be concluded that using Arabic Wikipedia as a KB to build a BOC for Arabic ATC achieves a higher accuracy in comparison with using Arabic WordNet. In addition, using Arabic Wikipedia concepts to build a BOC model to represent text in Arabic ATC performs better than BOW-LS as can be seen in Figure 6.18. The BOC based Wikipedia achieved the highest accuracy in comparison with various BOW models, which was more by 2.4%, 1.17%, and 1.5% with BOW-RE, BOW-LS, and BOW-Original respectively.

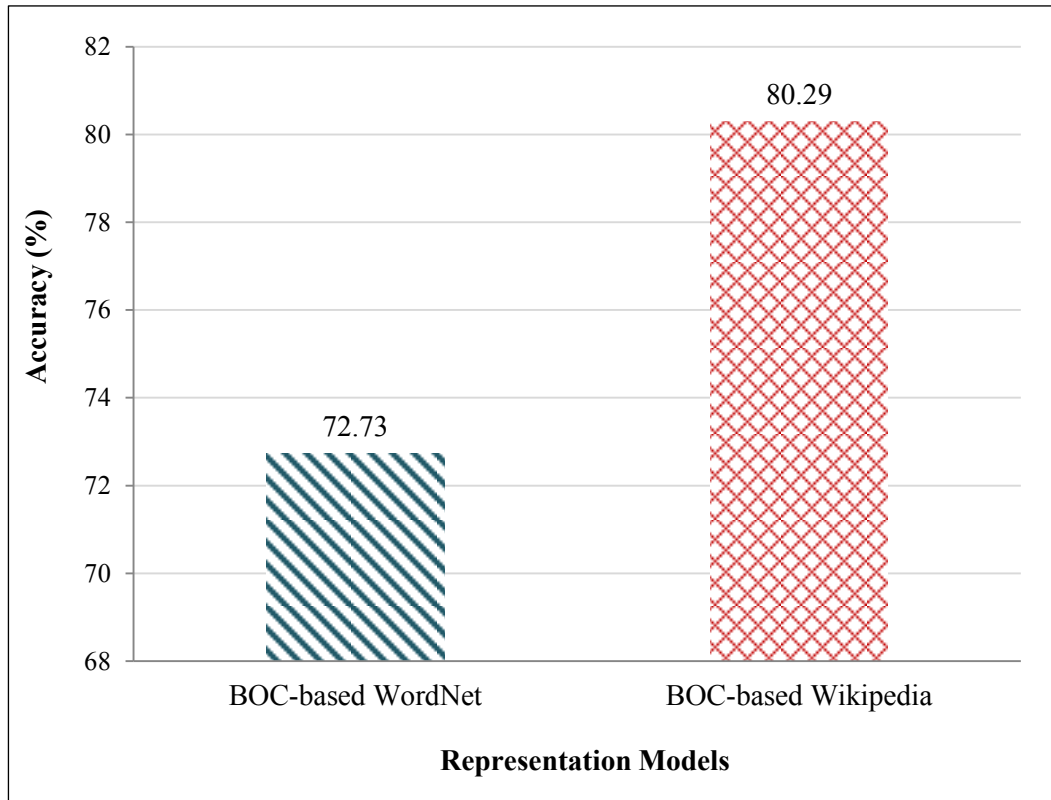


Figure 6.17 Average accuracy of all datasets with different classifiers

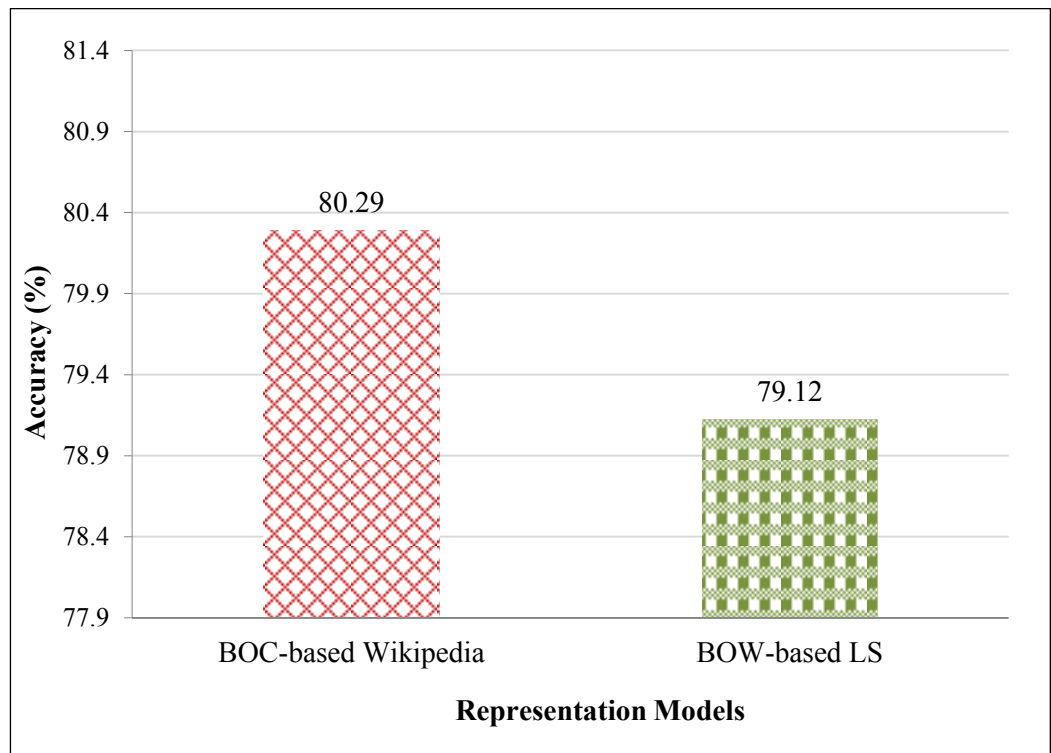


Figure 6.18 Average accuracy of BOW-LS and BOC-based Wikipedia in all datasets with different classifiers

Using a BOC-based Wikipedia or WordNet in an English ATC shows insufficient results compared to the BOW model. On the other hand, using Wikipedia concepts to represent Arabic text in ATC systems shows a better classification performance than the BOW model. Even though Arabic Wikipedia has only 5% of the number of concepts in comparison with English Wikipedia, this small amount of concepts shows a good ability to represent Arabic text in an ATC system. One of the reasons for this is the broad categories of the Arabic datasets used in this thesis, where all datasets contained documents from different newspapers and did not focus on specific topics. The other reason is the complex nature of the Arabic language and the poor morphological tools that makes Wikipedia concepts better features to represent text than words. For example, consider a text that discusses earth gravity ”جاذبية الأرض“ both words could appear in the text with different forms such as (جاذبيه الأرض), (جاذبية أرضية), (الجاذبية الأرضية). In the BOW model, the text is divided into single words such as (جاذبية), (الأرض), (جاذبيه), (الأرض), (جاذبية), (جاذبية), (الأرضية), (الجاذبية), (أرضية). Each word is stemmed in the pre-processing stage, and each stemmed method produced a different form of the words as can be seen in Table 6.13.

Table 6.13 root for the same words with different stemming method

Word	LS	RE	Real Expert
الجاذبيه	جاذبيه	جذب (pull)	جذب (pull)
جاذبيه	جاذبيه	جذب (pull)	جذب (pull)
الجاذبية	جاذبيه	جذب (pull)	جذب (pull)
جاذبية	جاذبيه	جذب (pull)	جذب (pull)
الأرض	أرض	أرض (Earth)	أرض (Earth)
الأرضيه	أرضيه	رضي (agree)	أرض (Earth)
الأرضية	أرضيه	رضي (agree)	أرض (Earth)
أرضيه	ارضيه	رضي (agree)	أرض (Earth)
أرضية	ارضيه	رضي (agree)	أرض (Earth)

From Table 6.13 it can be seen that both stemmed methods fail to get the correct form of the words. Both stemmed methods suffer from generating stemming

errors, especially where it is hard to distinguish between an extra letter and a root letter [208] such as (الأرضية). By using the LS method, we get four forms for the same two words where with RE method gets three forms for the same two words. On the other hand, Wikipedia concepts identified the different forms of the two words under one concept. In addition, most texts in Arabic are without diacritics (MSA format) which lead to additional ambiguity in the BOW model where two independent words in meaning stemmed to the same root such as (ارجل) “legs” and (رجال) “men”, stemmed to (رجل) which could be “leg” or “man” and there is no way to distinguish the correct meaning without the appearance of diacritics.

In an English ATC system, researchers combined both concepts and words using different strategies to solve the insufficient results from using concepts alone to represent text. These combined strategies are compared in Figure 6.19 where it can be seen that the ACC model achieved the highest accuracy compared to others with an average of 81.27% for all classifiers and datasets. In terms of classification accuracy, when used to represent text in ATC systems, the proposed CTD strategy to combine concepts and words is only lower than ACC and AC by 0.91 and 0.11 respectively, On the other hand, the CTD model has fewer features than ACC and AC which means a shorter time to train the classifier to predict a category for a text. In addition, the CTD model achieved a higher accuracy than both BOW-LS and BOC-based Wikipedia.

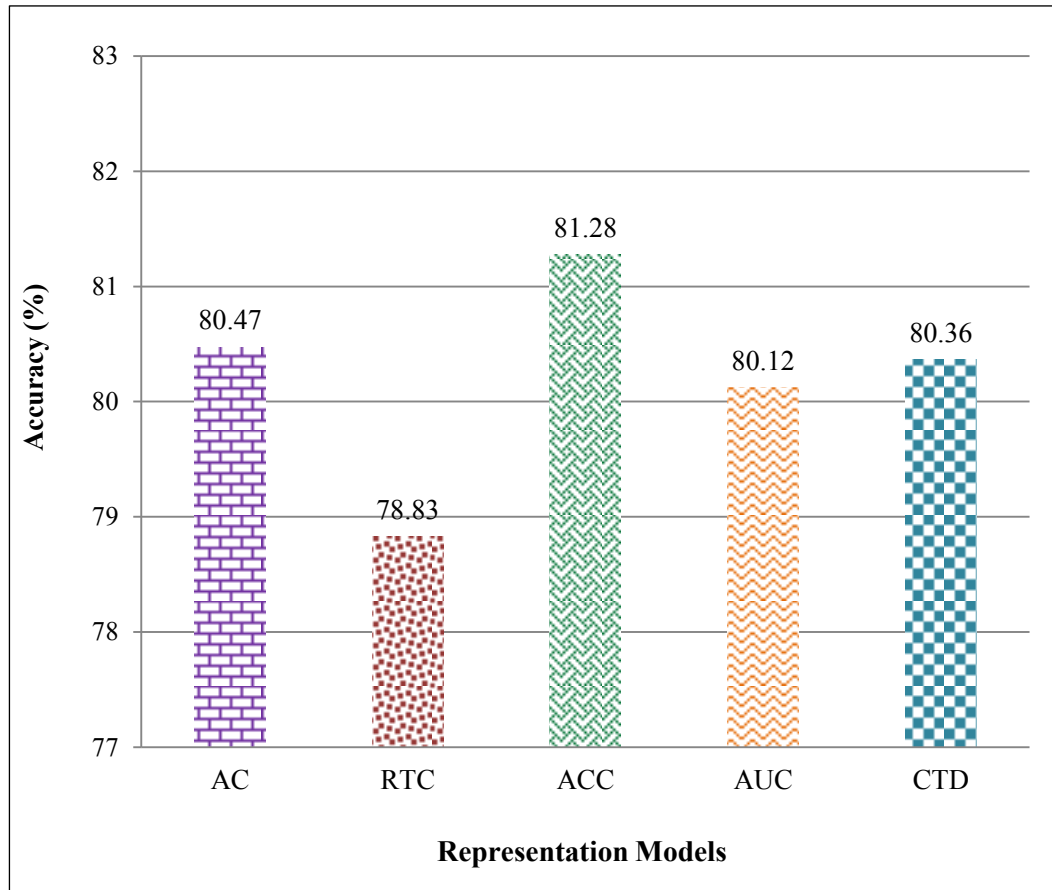


Figure 6.19 Average accuracy performance for combined representation models

On the whole, text representation models affect the performance of Arabic ATC not only in the accuracy of the system but also in the size of the model that affects the training time of the classification algorithms. Using Wikipedia concepts to represent Arabic text in an ATC system improves the performance of the TC task in comparison to using words, which is different than using concepts in an English ATC system. Using concepts also improves the performance of the system when accompanying it with words using the different strategies demonstrated earlier. Our proposed strategies work as efficiently as other proposed strategies. In addition, our proposed CTD strategy reduces the required time and memory for the classification system.

6.8 Summary

This chapter discussed the Arabic language and its complexity in comparison to other languages. The work that has been done for Arabic text in the area of ATC systems is limited in comparison with what has been done for English text. Most

of the reported studies focus on comparing different stemming methods, investigating the impact of pre-processing, applying different classification algorithms and comparing them on Arabic text in the ATC system. The most common studies on Arabic ATC systems are reviewed on the different points listed above.

To evaluate the proposed text representation strategies in the Arabic ATC system, three independent Arabic datasets are used. The datasets are diverse in the number of categories and documents; one of these datasets is a small dataset, and the other two are large publically available.

The experiment's design is described, and the experiment's results are reviewed and discussed. The impact of the morphological analysis is investigated in relation to the classification accuracy of the Arabic ATC system, the size of the representation models, and the time that is required by ML algorithms to train data and build a model to classify test data. In addition, both WordNet and Wikipedia were used as KBs to represent Arabic text using a BOC in an ATC system. The result shows Wikipedia has good coverage and provides rich information which makes it an ideal resource to provide concepts for Arabic text representation in an ATC system compared to WordNet. In addition, using Wikipedia concepts to represent Arabic text in ATC systems gives better classification accuracy than using words. The complex nature of the Arabic language and poor morphological tools make Wikipedia concepts better features to represent text.

In the field of Arabic text classification, previous reviews [209] show that only WordNet concepts are combined with words using the RTC strategy. Hence, Wikipedia concepts have not been used before to represent Arabic text in an ATC system. Different combined models that use both words and concepts from Wikipedia using different strategies (AC, RTC, ACC, AUC and CTD) that are first applied to Arabic text are represented and compared to each other. In terms of classification accuracy, the ACC is the best, followed by AC, CTD, and AUC. In addition, our CTD representation model offers more efficient implementation in terms of computation and memory requirements compared to other existing combined models that have been proposed to enrich text representation in ATC systems.

CHAPTER 7

Conclusions

7.1 Summary of thesis

This work is a study of Automatic Text Classification (ATC) which involves three components namely text pre-processing, text representation, and the classifier which is built using one of the ML algorithms. In specific, we investigated and assessed the effect of commonly available text representation models on the performance of ATC systems in regards to the accuracy of the classification and the efficiency of implementation. In addition, new models have been developed based on combining various selected aspects of the BOW and BOC representations in order to enhance the classification performance further.

Chapter 2 discussed the main components required to build an efficient ATC system. The chapter is concluded with a review of the different measures that evaluate the performance of an ATC system and the different methods used to divide a set of documents into training and testing subsets.

A detailed discussion of the text representation component which is an important aspect of a successful ATC system is covered in Chapter 3. We have outlined and discussed the advantages and disadvantages of different types of features namely single words, N-grams, statistical phrases, and syntactic phrases. Furthermore, we have considered concepts (which provide a unique meaning and behave as a unit of knowledge) used as features for text representation models. There is also a detailed description of the various characteristics of KBs and how concepts are determined in free text. This led to the selection of WordNet and Wikipedia as the KBs used in this thesis. In addition, the different strategies applied to use both words and concepts to represent text in ATC systems are discussed.

In Chapter 4 we have introduced two new strategies that use both words and concepts to enhance the classification performance, specifically, the accuracy of the classification and the efficiency of implementation. The first strategy is Adding Unmapped Concepts (AUC) and the second is Using Concepts for Terms which do not appear in the Document (CTD). For both developed text

representation strategies, the design of these proposed strategies is reviewed, and a document example is given to show how they enrich text representation in the VSM. We have compared the text representation models constructed using these proposed strategies to the basic BOW and BOC representation models and to other strategies previously used to combine words and concepts. The proposed strategies enriched the text representation for the selected documents in comparison to other representation models and improved the accuracy of the classification of the example documents.

In Chapter 5 we have built an experimental ATC system, evaluate its performance, and compare it to a number of different text representations as applied to three standard English datasets. First, we evaluate the performance of the system when the basic BOW model is used and compare that to when the basic BOC model is applied in conjunction with Wikipedia as a KB to provide concepts. This is then followed by similar evaluations when text representation approaches based on combining both words and concepts are used, as compared to when the basic BOW and BOC representations are used.

Finally, in Chapter 6 we have raised the question of the Arabic language and its complexity in comparison to other languages such as English. We have built an experimental ATC system, and its performance is evaluated and compared for a number of different text representations as applied to three standard Arabic datasets. The experiment design is clearly illustrated, and the impact of the morphological analysis is investigated in regards to the classification accuracy of the Arabic ATC system, the size of the representation models, and the time required by ML algorithms to train data and build a model to classify test data. In addition, concepts as features to represent Arabic text are used in the ATC system and both WordNet and Wikipedia as KBs are used to provide concepts. Both developed strategies and existing strategies that combined both words and concepts from Wikipedia are applied to the Arabic ATC system and compared to the basic BOW and BOC representations.

7.2 Main Findings & Contributions

The results of the experiments outlined in this thesis show that text representation is a key element influencing the performance of both English and Arabic ATC

systems, particularly in terms of classification accuracy. In order to further enhance the classification performance, we developed new models based on combining various selected aspects of the BOW and BOC representations. In this context, we developed two new approaches, the first is based on Adding Unmapped Concepts (AUC) and the second is based on using Concepts for Terms which do not appear in the Document (CTD). Both developed text representations demonstrated an improved classification performance in comparison to the use of the basic BOW and BOC representation models. They have also yielded comparable performance results to that obtained by the existing combined models investigated previously. In addition, our CTD representation model offers more efficient implementation in terms of computation and memory requirements compared to other existing combined models that have been proposed. Furthermore, commonly available ML algorithms are used with text representation models to assess the effect on the performance of ATC systems and investigate how different ML algorithms behave with different text representation models.

We discovered that when SVM, NB, and DT ML algorithms are used in the ATC system, the standard BOW model and the combined models achieved a higher classification accuracy than the BOC model. On the other hand, using the BOC model as a representation for English and Arabic text in the ATC system, in conjunction with the RF ML algorithm, showed better efficiency than the BOW.

Previous work conducted on Arabic text in the area of ATC systems is limited in comparison with what has been done for English text. Most of the reported studies focus on comparing different stemming methods, investigating the impact of pre-processing, applying different classification algorithms and comparing them on Arabic text in the ATC system. In addition, most of the datasets used in these works are in-house datasets, and where there are public datasets available it is limiting to use the same public dataset in more than one work under similar conditions. The unavailability of baseline datasets for comparing the accuracy of any newly developed Arabic ATC method with that of previously reported works is a disadvantage. Therefore, three datasets which are widely used in previous studies with different experimental setups is used. These datasets are diverse in the number of categories and documents that they contain. To evaluate our system with these datasets, a 10-fold cross-validation method is used to produce a baseline result for further comparison.

The impact of the morphological analysis was investigated in relation to the classification accuracy of the Arabic ATC system, the size of the representation models, and the time that is required by ML algorithms to train data and build a model to classify test data. A comparison between two KBs to represent Arabic text using a BOC model in an ATC system is evaluated. We discovered that Wikipedia is more effective than WordNet in general; Wikipedia has good coverage and provides rich information which makes it an ideal resource to provide concepts for Arabic text. In addition, using Wikipedia concepts to represent Arabic text in an ATC system gives better classification accuracy than using words. The complex nature of the Arabic language and poor morphological tools make Wikipedia concepts better features to represent text. For Arabic ATC systems, only WordNet concepts have been combined with words using an RTC strategy; Wikipedia concepts have not been used before. Different combined models that use both words and concepts from Wikipedia using different strategies (AC, RTC, ACC, AUC and CTD) are applied to Arabic text and compared to each other. Using Wikipedia concepts to enrich text representation offers an improvement over using the basic BOW/BOC models, as well as other reported combined models, for both English and Arabic ATC systems. Furthermore, our CTD representation model offers more efficient implementation in terms of computation and memory requirements compared to other existing combined models that have been proposed to enrich text representation in ATC systems.

In conclusion, each component in ATC system plays an important role in the classification accuracy. In addition, text representation not only influences the performance of ATC systems for both English and Arabic, it also impacts the effectiveness of ML algorithms.

7.3 Future work

In this work, we investigated available text representation models and their effect on the performance of ATC systems in general and concepts as units of knowledge that solve the problem of representing the semantic relationship between words in the text in particular. For this purpose two languages are used English and Arabic. Work with the Arabic language is still in the early stages and the following ideas may help direct future work:

- Improve morphological tools by proposing a method that uses Wikipedia concepts to reduce generating stemming errors such as in root extraction and light stemming methods.
- Use Wikipedia concepts to extend the representation model of short text such as social media.
- Investigate using Wikipedia concepts to represent more specific datasets than newspapers and discover if using Wikipedia concepts to build a BOC representation for Arabic ATC is still more efficiency than the BOW representation.

One general avenue of future research could investigate the Random Forest (RF) algorithm with other text representation features such as N- grams and syntactic phrase.

References

- [1] F. Peng and D. Schuurmans, "Combining naive bayes and n-gram language models for text classification," in *Proceedings of the 25th European conference on IR research*, 2003, pp. 335-350.
- [2] W. B. Cavnar and J. M. Trenkle, "N-gram-based text categorization," *Ann Arbor MI*, vol. 48113, pp. 161-175, 1994.
- [3] F. Peng, D. Schuurmans, and S. Wang, "Augmenting naive bayes classifiers with statistical language models," *Information Retrieval*, vol. 7, pp. 317-345, 2004.
- [4] B. Raskutti, H. Ferrá, and A. Kowalczyk, "Second order features for maximising text classification performance," in *Machine Learning: ECML 2001*, ed: Springer, 2001, pp. 419-430.
- [5] C. Sable, K. McKeown, and K. W. Church, "NLP found helpful (at least for one text categorization task)," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 2002, pp. 172-179.
- [6] R. Basili, A. Moschitti, and M. T. Pazienza, "Language sensitive text classification," in *RIAO*, 2000, pp. 331-343.
- [7] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann, 2005.
- [8] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, pp. 1-47, 2002.
- [9] M. E. Maron, "Automatic indexing: an experimental inquiry," *Journal of the ACM (JACM)*, vol. 8, pp. 404-417, 1961.
- [10] P. J. Hayes, P. M. Andersen, I. B. Nirenburg, and L. M. Schmandt, "Tcs: a shell for content-based text categorization," in *Artificial Intelligence Applications, 1990., Sixth Conference on*, 1990, pp. 320-326.
- [11] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, pp. 130-137, 1980.

- [12] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, pp. 613-620, 1975.
- [13] C. Silva and B. Ribeiro, "The importance of stop word removal on recall values in text categorization," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, 2003, pp. 1661-1666.
- [14] F. Sebastiani, "Classification of text, automatic," *The Encyclopedia of Language and Linguistics*, vol. 14, pp. 457-462, 2006.
- [15] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, pp. 513-523, 1988.
- [16] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, 2003.
- [17] T. Marill and D. M. Green, "On the effectiveness of receptors in recognition systems," *Information Theory, IEEE Transactions on*, vol. 9, pp. 11-17, 1963.
- [18] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *ICML*, 1997, pp. 412-420.
- [19] S. Li, R. Xia, C. Zong, and C.-R. Huang, "A framework of feature selection methods for text categorization," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, 2009, pp. 692-700.
- [20] T. M. Mitchell, "Machine learning. 1997," *Burr Ridge, IL: McGraw Hill*, vol. 45, 1997.
- [21] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational linguistics*, vol. 16, pp. 22-29, 1990.
- [22] Y. Yang, "An evaluation of statistical approaches to text categorization," *Information retrieval*, vol. 1, pp. 69-90, 1999.
- [23] T. Dunning, "Accurate methods for the statistics of surprise and coincidence," *Computational linguistics*, vol. 19, pp. 61-74, 1993.

- [24] V. Vapnic, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.
- [25] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features*: Springer, 1998.
- [26] S. Dumais and H. Chen, "Hierarchical classification of Web content," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000, pp. 256-263.
- [27] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *Proceedings of the seventh international conference on Information and knowledge management*, 1998, pp. 148-155.
- [28] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*: John Wiley & Sons, 2012.
- [29] H. Jiawei and M. Kamber, "Data mining: concepts and techniques," *San Francisco, CA, itd: Morgan Kaufmann*, vol. 5, 2001.
- [30] J. Han, M. Kamber, and J. Pei, "Data Mining: Concepts and Techniques, (The Morgan Kaufmann Series in Data Management Systems)," 2006.
- [31] A. Ben-Hur and J. Weston, "A user's guide to support vector machines," in *Data mining techniques for the life sciences*, ed: Springer, 2010, pp. 223-239.
- [32] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, pp. 121-167, 1998.
- [33] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," ed, 2003.
- [34] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871-1874, 2008.
- [35] I. Rish, "An empirical study of the naive Bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 2001, pp. 41-46.

- [36] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, 1998, pp. 41-48.
- [37] H. D. Margaret, "Data mining introductory and advanced topics," *LPE—Pearson Education Publisng*, 2003.
- [38] K.-M. Schneider, "On word frequency information and negative evidence in Naive Bayes text classification," *EsTAL*, vol. 3230, pp. 474-486, 2004.
- [39] J. Strickland, *Predictive Modeling and Analytics*: Lulu. com, 2014.
- [40] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 161-168.
- [41] D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," in *Machine learning: ECML-98*, ed: Springer, 1998, pp. 4-15.
- [42] H. Zhang, "The optimality of naive Bayes," *AA*, vol. 1, p. 3, 2004.
- [43] C. C. Aggarwal, *Data classification: algorithms and applications*: CRC Press, 2014.
- [44] O. Z. Maimon and L. Rokach, *Data mining and knowledge discovery handbook* vol. 1: Springer, 2005.
- [45] J. R. Quinlan, *C4. 5: programs for machine learning* vol. 1: Morgan kaufmann, 1993.
- [46] P.-N. Tan, M. Steinbach, and V. Kumar, "Classification: basic concepts, decision trees and model evaluation," *Introduction to data mining*, vol. 1, pp. 145-205, 2006.
- [47] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," ed, 2007.
- [48] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5-32, 2001.
- [49] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, pp. 123-140, 1996.

- [50] T. K. Ho, "The random subspace method for constructing decision forests," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, pp. 832-844, 1998.
- [51] R. Feldman and J. Sanger, *The text mining handbook: advanced approaches in analyzing unstructured data*: Cambridge University Press, 2007.
- [52] Z. Markov and D. T. Larose, *Data mining the Web: uncovering patterns in Web content, structure, and usage*: John Wiley & Sons, 2007.
- [53] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, 1995, pp. 1137-1145.
- [54] K. Aas and L. Eikvil, "Text categorisation: A survey," *Raport NR*, vol. 941, 1999.
- [55] B. Liu, "Supervised Learning," pp. 63-132, 2011.
- [56] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of*: Addison-Wesley, 1989.
- [57] C. C. Aggarwal and C. Zhai, "A survey of text clustering algorithms," in *Mining Text Data*, ed: Springer, 2012, pp. 77-128.
- [58] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 2002, pp. 79-86.
- [59] G. Armano and A. Giuliani, "Novel Text Summarization Techniques for Contextual Advertising," *Innovative Document Summarization Techniques: Revolutionizing Knowledge Understanding: Revolutionizing Knowledge Understanding*, p. 185, 2014.
- [60] G. Kanaan, R. Al-Shalabi, S. Ghwanmeh, and H. Al-Ma'adeed, "A comparison of text-classification techniques applied to Arabic text," *Journal of the American society for information science and technology*, vol. 60, pp. 1836-1844, 2009.
- [61] C. A. Martins, M. C. Monard, and E. T. Matsubara, "Reducing the dimensionality of bag-of-words text representation used by learning algorithms," in *Proceedings of The Third IASTED International Conference*

on Artificial Intelligence and Applications (AIA 2003), Benalmádena, Espanha.(to be published), 2003.

- [62] A. Kokawa, L. Busagala, W. Ohyama, T. Wakabayashi, and F. Kimura, "Feature Selection and Integration in Automatic Classification of Japanese Texts," in *Proceedings of the Second International Workshop on Regional Innovation Studies:(IWRIS2010)*, 2011, pp. 63-66.
- [63] S.-H. Lu, D.-A. Chiang, H.-C. Keh, and H.-H. Huang, "Chinese text classification by the Naïve Bayes Classifier and the associative classifier with multiple confidence threshold values," *Knowledge-based systems*, vol. 23, pp. 598-604, 2010.
- [64] P. Náther, "N-gram based Text Categorization," *Lomonosov Moscow State Univ*, 2005.
- [65] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, pp. 3-55, 2001.
- [66] E. Millar, D. Shen, J. Liu, and C. Nicholas, "Performance and scalability of a large-scale n-gram based information retrieval system," *Journal of digital information*, vol. 1, 2006.
- [67] G. K. Zipf, "Human behavior and the principle of least effort," 1949.
- [68] F. Peng and X. Huang, "Machine learning for Asian language text classification," *Journal of Documentation*, vol. 63, pp. 378-397, 2007.
- [69] M. Suzuki, N. Yamagishi, and Y.-C. Tsai, "Chinese text categorization using the character N-gram," in *Information Theory and its Applications (ISITA), 2012 International Symposium on*, 2012, pp. 722-726.
- [70] F. C. Ekmekcioglu, M. F. Lynch, and P. Willett, "Stemming and n-gram matching for term conflation in Turkish texts," *Information Research News*, vol. 7, pp. 2-6, 1996.
- [71] L. Khreisat, "A machine learning approach for Arabic text classification using N-gram frequency statistics," *Journal of Informetrics*, vol. 3, pp. 72-77, 2009.

- [72] A. Šilić, J.-H. Chauchat, B. D. Bašić, and A. Morin, "N-grams and morphological normalization in text classification: A comparison on a Croatian-English parallel corpus," in *Progress in Artificial Intelligence*, ed: Springer, 2007, pp. 671-682.
- [73] M. Suzuki, N. Yamagishi, Y.-C. Tsai, and M. Goto, "English and Japanese Text Categorization Using Word and Character N-grams," in *Proc. of the Asia Pacific Industrial Engineering & Management Systems Conference (APIEMS)*, 2012, pp. 715-722.
- [74] D. D. Lewis, "Representation and learning in information retrieval," University of Massachusetts, 1992.
- [75] D. D. Lewis and W. B. Croft, "Term clustering of syntactic phrases," in *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, 1989, pp. 385-404.
- [76] M. F. Caropreso, S. Matwin, and F. Sebastiani, "A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization," *Text databases and document management: Theory and practice*, pp. 78-102, 2001.
- [77] W. Zhang, T. Yoshida, and X. Tang, "Text classification based on multi-word with support vector machine," *Knowledge-Based Systems*, vol. 21, pp. 879-886, 2008.
- [78] K. Kageura and B. Umino, "Methods of automatic term recognition: A review," *Terminology*, vol. 3, pp. 259-289, 1996.
- [79] D. Mladenic and M. Grobelnik, "Word sequences as features in text-learning," in *In Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK98)*, 1998.
- [80] J. Furnkranz, "A study using n-gram features for text categorization," *Austrian Research Institute for Artificial Intelligence Technical Report OEFAI-TR-98-30 Schottengasse*, vol. 3, 1998.
- [81] D. Radošević, J. Dobša, D. Mladenic, Z. Stapić, and M. Novak, "Genre Document Classification Using Flexible Length Phrases," *Radošević, D., Dobša, J., Mladenić, D., Novak, M., Stapić, Z.:" Genre Document*

Classification Using Flexible Length Phrases", zbornik radova konferencije "Information and Intelligent Systems, 2006.

- [82] Britannica. *The Encyclopaedia Britannica*. Available: <http://www.britannica.com/>
- [83] Wikipedia. (2015). *Wikipedia, the free encyclopedia*. Available: https://en.wikipedia.org/wiki/Main_Page
- [84] J. Giles, "Internet encyclopaedias go head to head," *Nature*, vol. 438, pp. 900-901, 2005.
- [85] E. Gabrilovich and S. Markovitch, "Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge," in *AAAI*, 2006, pp. 1301-1306.
- [86] P. Wang and C. Domeniconi, "Building semantic kernels for text classification using wikipedia," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 713-721.
- [87] D. N. Milne, I. H. Witten, and D. M. Nichols, "A knowledge-based search engine powered by wikipedia," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 445-454.
- [88] M. Potthast, B. Stein, and M. Anderka, "A Wikipedia-based multilingual retrieval model," in *Advances in Information Retrieval*, ed: Springer, 2008, pp. 522-530.
- [89] S. Banerjee, K. Ramanathan, and A. Gupta, "Clustering short texts using wikipedia," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 787-788.
- [90] J. Hu, L. Fang, Y. Cao, H.-J. Zeng, H. Li, Q. Yang, *et al.*, "Enhancing text clustering by leveraging Wikipedia semantics," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 179-186.

- [91] E. Gabrilovich and S. Markovitch, "Wikipedia-based semantic interpretation for natural language processing," *Journal of Artificial Intelligence Research*, pp. 443-498, 2009.
- [92] M. Jarmasz, "Roget's thesaurus as a lexical resource for natural language processing," *arXiv preprint arXiv:1204.0140*, 2012.
- [93] S. Scott and S. Matwin, "Text classification using WordNet hypernyms," in *Use of WordNet in natural language processing systems: Proceedings of the conference*, 1998, pp. 38-44.
- [94] T. N. Mansuy and R. J. Hilderman, "Evaluating WordNet Features in Text Classification Models," 2006.
- [95] J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran, "Indexing with WordNet synsets can improve text retrieval," *arXiv preprint cmp-lg/9808002*, 1998.
- [96] E. M. Voorhees, "Using WordNet for text retrieval," *Fellbaum (Fellbaum, 1998)*, pp. 285-303, 1998.
- [97] M. Mohler and R. Mihalcea, "Text-to-text semantic similarity for automatic short answer grading," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, 2009, pp. 567-575.
- [98] G. Tsatsaronis, I. Varlamis, M. Vazirgiannis, and K. Nørnvåg, "Omiotis: A thesaurus-based measure of text relatedness," in *Machine Learning and Knowledge Discovery in Databases*, ed: Springer, 2009, pp. 742-745.
- [99] A. Hotho, S. Staab, and G. Stumme, "Wordnet improves Text Document Clustering," 2003.
- [100] D. R. Recupero, "A new unsupervised method for document clustering by using WordNet lexical and conceptual relations," *Information Retrieval*, vol. 10, pp. 563-579, 2007.
- [101] WordNet. *WordNet 3.0 database statistics* Available: <http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html>
- [102] ODP. (2006). *Open Directory Project*. Available: <https://www.dmoz.org/>

- [103] S. Zhu, J. Zeng, and H. Mamitsuka, "Enhancing MEDLINE document clustering by incorporating MeSH semantic similarity," *Bioinformatics*, vol. 25, pp. 1944-1951, 2009.
- [104] S. Bloehdorn and A. Hotho, "Boosting for text classification with semantic features," in *WebKDD*, 2004, pp. 149-166.
- [105] MeSH. (2003). *Medical subject headings (MeSH)*. National Library of Medicine. Available: <http://www.nlm.nih.gov/mesh>
- [106] AGROVOC. (1989). *AGROVOC Multilingual agricultural thesaurus* Available: <http://aims.fao.org/vest-registry/vocabularies/agrovoc-multilingual-agricultural-thesaurus>
- [107] W. Hersh, C. Buckley, T. Leone, and D. Hickam, "OHSUMED: An interactive retrieval evaluation and new large test collection for research," in *SIGIR '94*, 1994, pp. 192-201.
- [108] D. L. McCuinness, "Ontologies come of age," *Spinning the semantic web: bringing the World Wide Web to its full potential*, p. 171, 2005.
- [109] Y. Park, R. J. Byrd, and B. K. Boguraev, "Automatic glossary extraction: beyond terminology identification," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, 2002, pp. 1-7.
- [110] C. Fellbaum, "WordNet: An electronic lexical database. 1998," *WordNet is available from* <http://www.cogsci.princeton.edu/wn>, 2010.
- [111] BNC. (2007). *The British National Corpus*. Available: <http://www.natcorp.ox.ac.uk/>
- [112] G. Varelas, E. Voutsakis, P. Raftopoulou, E. G. Petrakis, and E. E. Milios, "Semantic similarity methods in wordNet and their application to information retrieval on the web," in *Proceedings of the 7th annual ACM international workshop on Web information and data management*, 2005, pp. 10-16.
- [113] M. Strube and S. P. Ponzetto, "WikiRelate! Computing semantic relatedness using Wikipedia," in *AAAI*, 2006, pp. 1419-1424.

- [114] R. Mihalcea and A. Csomai, "Wikify!: linking documents to encyclopedic knowledge," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 233-242.
- [115] D. Milne and I. H. Witten, "Learning to link with wikipedia," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 509-518.
- [116] M. d. B. Rodríguez, J. M. G. Hidalgo, and B. D. Agudo, "Using WordNet to complement training information in text categorization," *arXiv preprint cmp-lg/9709007*, 1997.
- [117] E. Brill, "A simple rule-based part of speech tagger," in *Proceedings of the workshop on Speech and Natural Language*, 1992, pp. 112-116.
- [118] D. D. Lewis. *The Reuters-21578 Dataset*. Available: <http://www.daviddlewis.com/resources/testcollections/reuters21578/>
- [119] C. Shaoul and C. Westbury. (2013, 4). *A reduced redundancy USENET corpus (2005–2011)*. Available: <http://www.psych.ualberta.ca/~westburylab/downloads/usenetcorpus.download.htm>
- [120] D. Bainbridge, C. G. Nevill-Manning, I. H. Witten, L. A. Smith, and R. J. McNab, "Towards a digital library of popular music," in *Proceedings of the fourth ACM conference on Digital libraries*, 1999, pp. 161-169.
- [121] M. Benkhalifa, A. Mouradi, and H. Bouyakhf, "Integrating external knowledge to supplement training data in semi-supervised learning for text categorization," *Information Retrieval*, vol. 4, pp. 91-113, 2001.
- [122] A. Kehagias, V. Petridis, V. G. Kaburlasos, and P. Fragkou, "A comparison of word-and sense-based text categorization using several classification algorithms," *Journal of Intelligent Information Systems*, vol. 21, pp. 227-247, 2003.
- [123] W. Francis and H. Kucera, "Frequency analysis of English usage," 1982.
- [124] E. Gabrilovich and S. Markovitch, "Feature generation for text categorization using world knowledge," in *IJCAI*, 2005, pp. 1048-1053.

- [125] P. Wang, J. Hu, H.-J. Zeng, L. Chen, and Z. Chen, "Improving Text Classification by Using Encyclopedia Knowledge," pp. 332-341, 2007.
- [126] P. Wang, J. Hu, H.-J. Zeng, and Z. Chen, "Using Wikipedia knowledge to improve text classification," *Knowledge and Information Systems*, vol. 19, pp. 265-281, 2008.
- [127] Volkan TUNALI. (2010 8 April). *Data Mining Research*. Available: <http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/>
- [128] A. Huang, "Similarity measures for text document clustering," in *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, Christchurch, New Zealand, 2008, pp. 49-56.
- [129] R. Maitra and I. P. Ramler, "A k-mean-directions algorithm for fast clustering of data on the sphere," *Journal of Computational and Graphical Statistics*, vol. 19, 2010.
- [130] J. C. Gomez and M.-F. Moens, "Document categorization based on minimum loss of reconstruction information," in *Advances in Computational Intelligence*, ed: Springer, 2013, pp. 91-103.
- [131] K. Lang. (13 April 2015). *The 20 Newsgroups data set*. Available: <http://qwone.com/~jason/20Newsgroups/>
- [132] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the 12th international conference on machine learning*, 1995, pp. 331-339.
- [133] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter, "Distributional word clusters vs. words for text categorization," *The Journal of Machine Learning Research*, vol. 3, pp. 1183-1208, 2003.
- [134] R. Ghani. (14 May 2015). *CMU World Wide Knowledge Base*. Available: <http://www.cs.cmu.edu/~webkb/>
- [135] M. Craven, A. McCallum, D. PiPasquo, T. Mitchell, and D. Freitag, "Learning to extract symbolic knowledge from the World Wide Web," DTIC Document1998.

- [136] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine learning*, vol. 39, pp. 103-134, 2000.
- [137] A. K. McCallum, "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering," ed, 1996.
- [138] A. M. d. A. Mesleh, "Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System," *Journal of Computer Science*, vol. 3, 2007.
- [139] D. Milne and I. H. Witten, "An open-source toolkit for mining Wikipedia," *Artificial Intelligence*, vol. 194, pp. 222-239, 2013.
- [140] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 42-49.
- [141] T. L. G. t. t. M. E. a. Africa. (2009, 5 May). Available: <http://www.melocalgate.com/lang.php>
- [142] S. A. Salama, *First Arabic Reader for beginners bilingual for speakers of English*: Language Practice Publishing.
- [143] H. Wehr, "A Dictionary of Modern Arabic," *Edited by J Milton Cowan*, vol. 3, p. 974, 1976.
- [144] K. Versteegh and C. Versteegh, *The Arabic Language*: Columbia University Press, 1997.
- [145] S. Khoja, "APT: Arabic part-of-speech tagger," in *Proceedings of the Student Workshop at NAACL*, 2001, pp. 20-25.
- [146] W. Wright and C. P. Caspari, *A Grammar of the Arabic Language*: Cosimo Classics, 2011.
- [147] C. H. M. Versteegh and K. Versteegh, *The Arabic Language*: Edinburgh University Press, 2014.
- [148] A. F. Nwesri, S. M. Tahaghoghi, and F. Scholer, "Capturing out-of-vocabulary words in Arabic text," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 2006, pp. 258-266.

- [149] M. Aljlal and O. Frieder, "On Arabic search: improving the retrieval effectiveness via a light stemming approach," in *Proceedings of the eleventh international conference on Information and knowledge management*, 2002, pp. 340-347.
- [150] S. Khoja, R. Garside, and G. Knowles, "A tagset for the morphosyntactic tagging of Arabic," *Computing Department Lancaster University*, 2001.
- [151] M. K. Saad and W. Ashour, "Arabic morphological tools for text mining," *Corpora*, vol. 18, p. 19, 2010.
- [152] M. Tayli and A. I. Al-Salamah, "Building bilingual microcomputer systems," *Communications of the ACM*, vol. 33, pp. 495-504, 1990.
- [153] B. Abuata and A. Al-Omari, "A rule-based stemmer for Arabic Gulf dialect," *Journal of King Saud University-Computer and Information Sciences*, vol. 27, pp. 104-112, 2015.
- [154] M. D. N. Habash, "Arabic Dialect Processing," 2009.
- [155] Z. Ibrahim, *Beyond lexical variation in modern standard Arabic: Egypt, Lebanon and Morocco*: Cambridge Scholars Publishing, 2009.
- [156] A. Mesleh and G. Kanaan, "Support vector machine text classification system: Using Ant Colony Optimization based feature subset selection," in *Computer Engineering & Systems, 2008. ICCES 2008. International Conference on*, 2008, pp. 143-148.
- [157] M. Abbas and K. Smaili, "Comparison of topic identification methods for arabic language," in *Proceedings of International Conference on Recent Advances in Natural Language Processing, RANLP*, 2005, pp. 14-17.
- [158] M. Abbas. (9 April 2015). *Arabic Corpora*. Available: <https://sites.google.com/site/mouradabbas9/corpora>
- [159] H. K. Chantar and D. W. Corne, "Feature subset selection for Arabic document categorization using BPSO-KNN," in *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, 2011, pp. 546-551.
- [160] H. Soori, M. Prilepok, J. Platos, E. Berhan, and V. Snasel, "Text Similarity Based on Data Compression in Arabic," in *AETA 2013: Recent Advances in*

Electrical Engineering and Related Sciences, ed: Springer, 2014, pp. 211-220.

- [161] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. Khorsheed, and A. Al-Rajeh, "Automatic Arabic text classification," 2008.
- [162] S. Alsaleem, "Automated Arabic Text Categorization Using SVM and NB," *Int. Arab J. e-Technol.*, vol. 2, pp. 124-128, 2011.
- [163] E. T. Al-Shammari, "Improving Arabic document categorization: Introducing local stem," in *Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on*, 2010, pp. 385-390.
- [164] Z. Elberrichi and K. Abidi, "Arabic Text Categorization: a Comparative Study of Different Representation Modes," *International Arab Journal of Information Technology (IAJIT)*, vol. 9, 2012.
- [165] A. M. Mesleh, "Feature sub-set selection metrics for Arabic text classification," *Pattern Recognition Letters*, vol. 32, pp. 1922-1929, 2011.
- [166] F. Ciravegna, L. Gilardoni, A. Lavelli, S. Mazza, W. J. Black, M. Ferraro, *et al.*, "Flexible text classification for financial applications: the FACILE system," in *ECAI*, 2000, pp. 696-700.
- [167] D. Yarowsky, "A comparison of corpus-based techniques for restoring accents in Spanish and French text," in *Natural language processing using very large corpora*, ed: Springer, 1999, pp. 99-120.
- [168] D. Yarowsky, "Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, 1994, pp. 88-95.
- [169] S. Spat, B. Cadonna, I. Rakovac, C. Gütl, H. Leitner, G. Stark, *et al.*, "Enhanced information retrieval from narrative German-language clinical text documents using automated document classification," *Studies in health technology and informatics*, vol. 136, pp. 473-8, 2008.
- [170] F. Peng, X. Huang, D. Schuurmans, and S. Wang, "Text classification in Asian languages without word segmentation," in *Proceedings of the sixth international workshop on Information retrieval with Asian languages- Volume 11*, 2003, pp. 41-48.

- [171] Z.-l. GONG, D.-x. ZHANG, and M.-m. HU, "An Improved SVM Algorithm for Chinese Text Classification [J]," *Computer Simulation*, vol. 7, p. 040, 2009.
- [172] T. Zagibalov and J. Carroll, "Automatic seed word selection for unsupervised sentiment classification of Chinese text," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, 2008, pp. 1073-1080.
- [173] N. Hui and L. Zhi-long, "A Study on Feature Selection in Chinese Text Categorization [J]," *Computer Knowledge and Technology (Academic Exchange)*, vol. 21, p. 076, 2007.
- [174] J. G. C. Association, "Japanese classification of gastric carcinoma—2nd English edition—," *Gastric cancer*, vol. 1, pp. 10-24, 1998.
- [175] D. B. Bracewell, J. Yan, F. Ren, and S. Kuroiwa, "Category classification and topic discovery of japanese and english news articles," *Electronic Notes in Theoretical Computer Science*, vol. 225, pp. 51-65, 2009.
- [176] F. Harrag, E. El-Qawasmah, and A. M. S. Al-Salman, "Stemming as a feature reduction technique for Arabic Text Categorization," in *Programming and Systems (ISPS), 2011 10th International Symposium on*, 2011, pp. 128-133.
- [177] M. M. Syiam, Z. T. Fayed, and M. B. Habib, "An intelligent system for Arabic text categorization," *International Journal of Intelligent Computing and Information Sciences*, vol. 6, pp. 1-19, 2006.
- [178] K. Darwish and D. W. Oard, "Adapting Morphology for Arabic Information Retrieval*," in *Arabic Computational Morphology*, ed: Springer, 2007, pp. 245-262.
- [179] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002, pp. 275-282.
- [180] C. Sakhr. (1997). *The Encyclopedia of the Nine Books for the Honorable Prophetic Traditions*. Available: <http://www.Harf.com>

- [181] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Light stemming for Arabic information retrieval," in *Arabic computational morphology*, ed: Springer, 2007, pp. 221-243.
- [182] S. Khoja and R. Garside, "Stemming arabic text," *Lancaster, UK, Computing Department, Lancaster University*, 1999.
- [183] R. Duwairi, M. N. Al-Refai, and N. Khasawneh, "Feature reduction techniques for Arabic text categorization," *Journal of the American society for information science and technology*, vol. 60, pp. 2347-2352, 2009.
- [184] H. M. Al-Serhan, R. Al Shalabi, and G. Kannan, "New approach for extracting Arabic roots," 2003.
- [185] M. El Kourdi, A. Bensaid, and T.-e. Rachidi, "Automatic Arabic document categorization based on the Naïve Bayes algorithm," in *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, 2004, pp. 51-58.
- [186] M. El Kourdi, T. Rachidi, and A. Bensaid, "A concatenative approach to Arabic word root extraction," ed: progress, 2004.
- [187] G. Kanaan, R. Al-Shalabi, and A. Al-Akhras, "kNN Arabic text categorization using IG feature selection," in *Proceedings of The 4th International Multiconference on Computer Science and Information Technology*, 2006, pp. 5-7.
- [188] F. Harrag, E. El-Qawasmeh, and P. Pichappan, "Improving Arabic text categorization using decision trees," in *Networked Digital Technologies, 2009. NDT'09. First International Conference on*, 2009, pp. 110-115.
- [189] L. Galavotti, F. Sebastiani, and M. Simi, "Experiments on the use of feature selection and negative evidence in automated text categorization," in *Research and Advanced Technology for Digital Libraries*, ed: Springer, 2000, pp. 59-68.
- [190] Z. Zheng, R. Srihari, and S. Srihari, "A feature selection framework for text filtering," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, 2003, pp. 705-708.

- [191] H. T. Ng, W. B. Goh, and K. L. Low, "Feature selection, perceptron learning, and a usability case study for text categorization," in *ACM SIGIR Forum*, 1997, pp. 67-73.
- [192] D. Mladenic and M. Grobelnik, "Feature selection for unbalanced class distribution and naive bayes," in *ICML*, 1999, pp. 258-267.
- [193] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *The Journal of machine learning research*, vol. 3, pp. 1289-1305, 2003.
- [194] R. Al-Shalabi and M. Evens, "A computational morphology system for Arabic," in *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, 1998, pp. 66-72.
- [195] A. Chen and F. C. Gey, "Building an Arabic Stemmer for Information Retrieval," in *TREC*, 2002, pp. 631-639.
- [196] R. M. Duwairi, "Arabic Text Categorization," *Int. Arab J. Inf. Technol.*, vol. 4, pp. 125-132, 2007.
- [197] S. Al-Saleem, "Associative classification to categorize Arabic data sets," *The International Journal Of ACM JORDAN*, vol. 1, pp. 118-127, 2010.
- [198] I. Hmeidi, B. Hawashin, and E. El-Qawasmeh, "Performance of KNN and SVM classifiers on full word Arabic articles," *Advanced Engineering Informatics*, vol. 22, pp. 106-111, 2008.
- [199] W. e. M. Hadi, M. a. Salam, and J. A. Al-Widian, "Performance of nb and svm classifiers in islamic arabic data," in *Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications*, 2010, p. 14.
- [200] M. Benkhalifa, A. Mouradi, and H. Bouyakhf, "Integrating WordNet knowledge to supplement training data in semi-supervised agglomerative hierarchical clustering for text categorization," *International Journal of Intelligent Systems*, vol. 16, pp. 929-947, 2001.
- [201] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "An kNN model-based approach and its application in text categorization," in *Computational Linguistics and Intelligent Text Processing*, ed: Springer, 2004, pp. 559-570.

- [202] A. Samir, W. Ata, and N. Darwish, "A new technique for automatic text categorization for Arabic documents," in *5th IBIMA Conference (The internet & information technology in modern organizations)*, 2005.
- [203] R. Al-Shalabi and R. Obeidat, "Improving KNN Arabic text classification with n-grams based document indexing," in *Proceedings of the Sixth International Conference on Informatics and Systems, Cairo, Egypt*, 2008, pp. 108-112.
- [204] stop-words. (20 Nov 2011, 10 April 2014). *Arabic Stop words* [Online]. Available: <https://code.google.com/p/stop-words/source/browse/trunk/stop-words/stop-words/stop-words-arabic.txt?r=3>
- [205] A. Browser. *Arabic WordNet Browser*. Available: <http://globalwordnet.org/arabic-wordnet/awn-browser/>
- [206] *Arabic Wikipedia XML dump* [Online]. Available: <http://dumps.wikimedia.org/arwiki/>
- [207] M. K. Saad, "The impact of text preprocessing and term weighting on Arabic text classification," The Islamic University-Gaza, 2010.
- [208] E. T. Al-Shammari and J. Lin, "Towards an error-free Arabic stemming," in *Proceedings of the 2nd ACM workshop on Improving non english web searching*, 2008, pp. 9-16.
- [209] A. Karima, E. Zakaria, T. G. Yamina, A. A. S. MOHAMMED, R. P. SELVAM, V. VENKATAKRISHNAN, *et al.*, "Arabic text categorization: a comparative study of different representation modes," *Journal of Theoretical and Applied Information Technology*, vol. 38, pp. 1-5, 2012.

Appendix A

Documents Sample

Document 1 *The Magical Number Seven, Plus or Minus Two:***cisi.001232***Some Limits on Our Capacity for Processing Information*

My problem is that I have been persecuted by an integer. For seven years this number has followed me around, has intruded in my most private data, and has assaulted me from the pages of our most public journals. This number assumes a variety of disguises, being sometimes a little larger and sometimes a little smaller than usual, but never changing so much as to be unrecognizable. The persistence with which this number plagues me is far more than a random accident. There is, to quote a famous senator, a design behind it, some pattern governing its appearances. Either there really is something unusual about the number or else I am suffering from delusions of persecution.

Document 2***Current Awareness Publications An Evaluation*****cisi.001363**

Three types of current awareness publications are defined in terms of their generation and distribution characteristics. These types are the SDI type, the intermediate type such as NASA's SCAN program, and the unselected type such as the usual library accession list. A fourth type, the indexing and abstracting journal is defined but not related to the other three types. The three current awareness publications are shown to have a relationship in terms of the number of requests they can produce from a given set of references or notices. The SDI system will produce about one request from ten notices; the intermediate type about one request per hundred notices; and the unselected type, one request per thousand notices. Despite the differences in the number of notices to generate one request, the cost to produce one request is about the same for the three types of tools. Also, the usage of these tools is related to the amount of user time available and a curve is presented based on various assumptions to define this relationship.

Document 3 *The Review of Scientific Instruments with Physics News and
cisi.001260 Views*

A study of the periodicals for the subjects of physic and radio has been undertaken with the hope of indicating those which are most used in each field. The method selected is similar to that used by P.L.K. Gross and E.M. Gross.

Document 4 *Damage incurred on a tilt-wing multipropeller vtol/stol aircraft*
cran.001168 *operating over a level, gravel-covered surface .*

a summary is presented of the damage experienced by a tilt-wing vtol stol aircraft as a result of operatin from a level surface covered with loose gravel . the damage was inadvertently incurred as the aircraft was performing a taxiing-turn maneuver over an area of level macadam surface containing loose and embedded crushed stone . observers from a chase aircraft commented that the wing was tilted at approximately 76 with respect to the ground when the damage occurred . deposits of stone in the open fuselage, punctures in the rotor blade skin, and damage to the compressor blades of the engine occurred due to th circulation of the crushed gravel .

Document 5 *Application of inequality constraints to variational*
cran.001217 *problems of lifting, no centre, inequality constraints, one instead*

problems of lifting re-entry . inequality constraints are introduced into the variational formulation of the optimum re-entry problem for a lifting vehicle to prevent human and or structural tolerances from being exceeded . these constraints consist of minimum and maximum angle of attack, maximum load factor, and maximum convective heat transfer (equilibrium temperature) . the equations have been programmed for the ibm 704 computer, and sample trajectories are presented for which the total heat transferred to certain critical areas on the windward surface of the vehicle is minimized . these trajectories indicate the dominant effect of the constraints on the optimum flight path, which is shown to consist of both unconstrained and constrained arcs .

Document 6	<i>Design of stiffened cylinders in axial compression</i>
cran.001293	<i>the problem of optimum design of axially compressed cylinders stiffened by rings and stringers is discussed . particular attention is devoted to configurations suitable for large launch vehicles . consideration is given to the analytical techniques for determining strength as well as the procedures for optimization .</i>
Document 7	<i>mansonias crassipes as the natural vector of filarioids, plasmodium gallinaceum and other plasmodia of fowls in Ceylon it would appear from these observations that m. crassipes is the natural vector of p. gallinaceum of fowls in ceylon and possibly of other species of plasmodium of fowls or of related avia hosts. if one of the other plasmodia from m. crassipes is in fact p. lophurae then this will be the first record of p. lophurae from this part of the world. we are, however, not certain whether the fowl is the natural vertebrate host of p. lophurae in ceylon. we have seen crithidial forms in the midgut of engorged wild-caught m. crassipes. This mosquito may, therefore, prove to be the vector of the trypanosome as well. The trypanosome differs from trypanosoma gallinarum and t. calmettei which have previously been reported from fowls.</i>
Document 8	<i>autopathy: follow-up of cases the publication of two papers recently in the acta paedopsychiatrica by d. arn. van krevelen on the subject of "autistic psychopathy", has stimulated the author to present 5 cases which appear to conform to his description. the only value of this brief contribution lies in the fact that they have all been followed up for at least 10 years, and this may give some idea of the eventual results in this type of cases.</i>

Document 9 *clinical experience with palliation of metastatic adenocarcinoma*
med.000777 *with 5-fluorouracil chemotherapy chemotherapy of solid tumors is*
 becoming a more and more helpful therapeutic tool.

It is probable that within a few years chemotherapy will become firmly established as an adjunct to the primary treatment of solid tumors. perhaps someday a cancerocidal chemical will be devised that can actually act as antibiotics do on bacteria. for this reason it is important for all surgeons interested in the treatment of cancer to try to keep abreast of the recent advances in the field of chemotherapy. Surgeons should also begin to use some of the technics for palliation of incurable patients so that when these drugs are added to the armamentarium of primary treatment of cancer, they will be familiar with the various technics and the mode of actions of the drugs. 5-fluorouracil (5-fu) has had extensive clinical use and has produced significant palliation in breast and colon cancers with about a third of these patients receiving palliation for five to six months.

Appendix B

Papers Publications

Arabic Text Classification using Bag-of-Concepts Representation

Alaa Alahmadi, Arash Joorabchi and Abdulhussain E. Mahdi
Electronic and Computer Engineering Department, University of Limerick, Ireland
{alaa.alahmadi; arash.joorabchi; hussain.mahdi}@ul.ie

Keywords: Automatic text classification; Arabic text; bag-of-words; bag-of-concepts; Wikipedia.

Abstract: With the exponential growth of Arabic text in digital form, the need for efficient organization, navigation and browsing of large amounts of documents in Arabic has increased. Text Classification (TC) is one of the important subfields of data mining. The Bag-of-Words (BOW) representation model, which is the traditional way to represent text for TC, only takes into account the frequency of term occurrence within a document. Therefore, it ignores important semantic relationships between terms and treats synonymous words independently. In order to address this problem, this paper describes the application of a Bag-of-Concepts (BOC) text representation model for Arabic text. The proposed model is based on utilizing the Arabic Wikipedia as a knowledge base for concept detection. The BOC model is used to generate a Vector Space Model, which in turn is fed into a classifier to categorize a collection of Arabic text documents. Two different machine-learning based classifiers have been deployed to evaluate the effectiveness of the proposed model in comparison to the traditional BOW model. The results of our experiment show that the proposed BOC model achieves an improved performance with respect to BOW in terms of classification accuracy.

1 INTRODUCTION

Arabic, which is the fifth most spoken language in the world, belongs to the Semitic group of languages and it is a highly inflectional and derivational language (Versteegh and Versteegh, 1997). The Arabic alphabet consists of 28 letters, and letters are directly connected when forming words. Arabic text is read from right to left. Unlike English, proper nouns do not start with capital letters which makes the process of recognizing and extracting them difficult. Moreover, 22 of the letters take different shapes based on their position in the word, i.e., initial, medial, or final. Table 1 shows the letter *b* (ب) appearing in different sample words and how its position affects its shape.

Table 1: Example of an Arabic letter and its various shapes depending on its position in the word.

position	word	Letter shape
Start	بدر	بـ
Middle	قبر	ـبـ
End	قلب	ب

The exponential growth of Arabic documents in digital form on the web has increased the need to assist users with the fast and effective navigation, browsing, and discovery of useful information on the Internet. Text Classification (TC) is the task of assigning one or more predefined categories to a given text.

Classification of Arabic text can be a challenging task due to the rich and complex nature of the Arabic language. The majority of reported works on Arabic TC attempt to represent text by using the Bag-of-Words (BOW) model. In this model, each document, $d_i \in D$, is expressed as a weighted high dimensional vector, $\vec{d}_i$, where each dimension corresponds to a unique word. For example, Alsaleem (2011) used Support Vector Machine (SVM) and Naïve Bayes (NB) classification algorithms using the BOW model to classify a Saudi Arabic newspaper text collection by Al-Harbi et al. (2008). The classification system yielded a Macro F1 score of 77.85% and 74.0% for SVM and NB respectively. Kanaan et al. (2009) used the BOW model for Arabic TC using a dataset compiled by Mesleh (2007) with different

weighting schemes such as Term Frequency (TF), Term Frequency Inverse Document Frequency (TFIDF), and Weighted Inverse Document Frequency (WIDF). Performance comparisons between k -Nearest Neighbors (k -NN), Rocchio, and NB as classifiers were conducted. Results of this experiment show that the WIDF scheme yields the best performance when used in conjunction with the k -NN, while TFIDF shows the best performance when used in conjunction with Rocchio. Among the three classifiers, the NB classifier is reported to be the best performer yielding a Macro F1 score of 84.53%. Mesleh (2007) reported a BOW based Arabic TC system which uses Chi-square for feature selection. Their results show that using a SVM classifier in this context yields better classification performance compared to a k -NN or a NB classifier when features are reduced using Chi-square. It yields a Macro F1 score of 88.11%, when evaluated using their in-house compiled Arabic text dataset.

The BOW model suffers from two main limitations: (1) it breaks terms into their constituent words, e.g., it breaks 'text classification' into the words 'text' and 'classification'; as a result the order of the words is lost in the model and the meaning of the terms could be changed; (2) it treats synonymous words as independent features, e.g., 'classification' and 'categorization' are considered as two independent words with no semantic association. As a result of that documents which discuss similar topics and contain synonymous words could be considered unrelated.

Researchers have attempted to address the above issues in English TC by representing text as concepts rather than words, using an approach known as Bag-of-Concepts (BOC). A concept is a unit of knowledge with a unique meaning (ISO, 2009). To build a BOC model, semantic knowledge bases such as WordNet¹, Open Directory Project (ODP)², and Wikipedia³ are used to identify the concepts appearing within a document. By using concepts in text representation the semantics and associations between words appearing in the document will be preserved. For example, Hotho et al (2003) used the English Wordnet as a knowledge base to represent English text. For each term in a document, Wordnet returns an ordered list of synonyms and the first ranked synonym will be used as a concept for the term. In that study, three approaches were proposed for using concepts as

features for text representation; (1) using only concepts to represent documents; (2) Adding Concept (AC) as complimentary features to the BOW model ; (3) Replacing Term words with Concepts (RTC) in BOW model. The study shows that AC yields better classification performance results than the other two approaches. The results of this study show that representing documents with only concepts is insufficient as WordNet does not cover all special domain vocabularies. Furthermore, WordNet is limited as it is a manually constructed dictionary and therefore laborious to maintain.

To deal with this problem, other researchers have tried replacing WordNet with other knowledge bases derived from the Internet, such as ODP and Wikipedia, e.g., see (Gabrilovich and Markovitch, 2005, Gabrilovich and Markovitch, 2006). In these studies, ODP categories and Wikipedia articles are used as concepts for text representation. For each document, a text fragment (such as word, sentence, paragraph, or the whole document) maps to the most relevant ODP categories or Wikipedia articles. The mapped concepts are added to the document using the AC approach. Using these knowledge bases for BOC modelling improved the performance when applied to English TC compared to BOW model.

For Arabic TC, Elberrichi and Abidi (2012) used the Arabic WordNet (Black et al., 2006) to identify concepts appearing within the documents. A comparison between different text representation models such as BOW, N-grams and BOC was conducted using an Arabic text dataset collected by Mesleh (2007). An RTC variation of BOC used in conjunction with Chi-square for feature selection and a k -NN classifier was reported to achieve higher performance results compared to other representations.

In a previous work, we developed a number of new approaches, which combine the BOW and the BOC models, and applied them to English TC (Alahmadi et al., 2013) and Arabic TC (Alahmadi et al., 2014). In (Alahmadi et al., 2014), a NB classification algorithm was shown to provide better performance in conjunction with the BOC model compared to BOW model. This current work focuses on this point. The remainder of the paper is organized as follows: Section 2 describes why Wikipedia is a suitable knowledge base for BOC modelling of Arabic text. Section 3 outlines the pre-processing phase and describes the BOC model in details. The experimental set-up and results are discussed in Section 4. The paper concludes in Section 5.

¹ <http://wordnet.princeton.edu/>.

² <http://dmoz.org>.

³ <http://www.wikipedia.org>.

2 WIKIPEDIA

Wikipedia is the largest electronic knowledge repository on the Internet and its content is entirely contributed collaboratively by volunteers. Wikipedia is a comprehensive, up to date, and well-formed knowledge source.

In order to use Wikipedia as a knowledge source for building BOC models of Arabic text, an open-source toolkit known as Wikipedia-Miner (Milne and Witten, 2013) is utilized. First the toolkit processes Wikipedia XML dump files⁴ and creates a database that contains a summarized version of Wikipedia's content and structure. The toolkit assigns each Wikipedia page a unique id, title, and type (article, category, or redirect). Each article in Wikipedia describes a single concept and the article title is the descriptor of the concept. The title is well formed, brief and can be used as a descriptor in ontologies. The aim of the redirect pages in Wikipedia is to connect articles with alternative titles that correspond to their synonyms.

The great majority of Wikipedia articles, i.e. concepts, are classified to one or more Wikipedia categories, which are hierarchical and descend from a single root. The maximum category depth in Arabic Wikipedia is 12. A concept can belong to multiple parent categories which are more general articles than the concept itself. In addition more specific concepts which are child articles can be mined by the concept as a parent category for them.

In Wikipedia, an article may have more than one redirect and other Wikipedia articles may link to it (link anchors). All these elements offer additional information, and they are grouped into labels. These labels will be used as concepts to represent text in different applications. In this study we have used Wikipedia labels to build the BOC representation model for Arabic TC.

3 ARABIC TEXT CLASSIFICATION WITH BOC MODEL

This section describes the developed TC system as shown in Figure 1.

The process begins by passing all the documents in the dataset through a text pre-processing phase where they are cleaned and processed. The result of

this phase is a set of well-defined features. These features are then used in the modeling phase. Based on the features, i.e. words or concepts, two independent representation models are built; namely the BOW and BOC. Finally, the outputs of these phases are fed to two distinct Machine Learning (ML) based classification algorithms (classifiers). These algorithms learn from the labeled training texts to predict the class of unlabeled testing texts.

The remaining of this section elaborates on each of these phases.

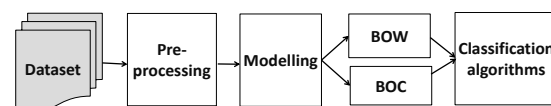


Figure 1: Developed TC system

3.1 Text Pre-processing

The pre-processing is the task of converting text to a well-defined set of features. In this phase all noise and irregularities, which negatively affect the classification performance, will be removed.

This phase includes the following steps:

- All digits, punctuation marks, and non-Arabic characters are removed.
- For normalization we follow (Kanaan et al., 2009, Mesleh, 2007) and remove diacritics and normalize some Arabic letters as follows:
 - Normalization of (ل) by replacing (ل), (ل) and (ل) at the start of words with (ل).
 - Normalization of (ي) by replacing (ي) at the end of words with (ي).
 - Normalization of (ة) by replacing (ة) at the end of words with (ة).
- Arabic stop words and common words such as pronouns and prepositions are removed from text as they do not carry any discriminatory significance so far classification is concerned.
- There are two slightly different approaches to stemming in Arabic language: (a) Root Extraction (RE) in which a set of prefixes, suffixes, and infixes are removed from words to extract root; and (b) Light Stemming (LS) in which only prefixes and suffixes are removed from words. Prefixes in Arabic words are groups of letters added to the beginning of the root, these letters could be definite articles, prepositions, pronouns and connectives, such as (ال), (وال), (بال), (كال), (فال). Suffixes in Arabic words are set of letters that are added to the end of the root, such as (ة), (ات), (هـ), (ان), (ون), (يه) and (ي). Infixes in Arabic words are

⁴ <http://dumps.wikimedia.org/arwiki/>

set of letters that inserted into root to create new words; such as (ل), (ت), (و), (ن) and (ي).

In this work, LS approach has been used as it has been found to yield better results compared to RE for Arabic TC (Kanaan et al., 2009, Mesleh, 2007).

- e. Remove words that occur less than 4 times in the dataset to reduce the dimensionality of the feature vector of the representation model (Mesleh, 2007).

3.2 Text Representation

The result of pre-processing phase is a set of features that represent Arabic text. In our classification system features can be either words or concepts and, therefore two independent representation models are created, i.e., BOW and BOC. By using words as features, a BOW model is created and each word is weighted using the TFIDF weighting scheme (Salton and Buckley, 1988):

$$TFIDF(w_m, d_i) = TF(w_m, d_i) \cdot IDF(w_m) \quad (1)$$

where $TF(w_m, d_i)$ is the frequency of a word, $w_m \in W$, W is the set of all words that are considered as features.

$$TF(w_m, d_i) = \frac{n(w_m, d_i)}{|d_i|} \quad (2)$$

where $n(w_m, d_i)$ is the occurrence frequency of the word w_m in document d_i , normalized by $|d_i|$ which is the length of d_i . The inverse document frequency $IDF(w_m)$ is defined as:

$$IDF(w_m) = \log_e \frac{|D|}{DF(w_m)} \quad (3)$$

where $DF(w_m)$ returns document frequency of the word w_m . It counts the number of documents in D where the word w_m appears, $|D|$ is the total number of documents in the dataset. The $IDF(w_m)$ parameter has the effect of reducing the weight of those words which appear in a large portion of the dataset D .

To create the BOC model, concepts within a given text need to be identified using the Wikipedia-Miner toolkit. First, a document is processed and cleaned. Then, all possible Wikipedia concepts in the document will be identified using the topic detection functionality in Wikipedia-Miner. A majority of detected concepts are ambiguous as they may refer to multiple meanings.

The disambiguation component of the toolkit is used to compute a probability estimate for each potential concept based on its relevance to other detected concepts in the document. The probability determines if a concept is relevant to the context or not and the ambiguity will be resolved. A set of candidate concepts are selected which are used to represent documents in the BOC model.

For two given documents d_1 and d_2 which belong to SNP dataset (see Section 4 for the details of this dataset); the first document belongs to 'Sport' category and describes a news article about Real Madrid football team, whereas the other document belongs to the 'Economic' category and discusses the Saudi stock market. The word Riyal (ريال) is appeared in both documents but with different intended semantic meaning. In d_1 it refers to Real Madrid (ريال مدريد) team and in d_2 it refers to Saudi Riyal (ريال سعودي) currency. By using BOW representation the difference in the meanings are lost and the Riyal (ريال) is considered the same in both documents. Furthermore, as can be seen in Table 2, the value for Riyal (ريال) document frequency in the dataset is 267 which reflects the generality of this feature. This leads to reducing the TFIDF weight of this feature and as a result the document is classified to a wrong category.

Table 2: Sample of features from SNP dataset and their Document Frequency (DF) values

DF	Features				
	Riyal (ريال)	Madrid (مدريد)	Saudi (سعودي)	Saudi Riyal (ريال سعودي)	Real Madrid (ريال مدريد)
	267	23	581	179	9

However, by using the BOC model various meanings of the terms will be identified and mapped to their corresponding concepts, e.g., Riyal (ريال) such as Saudi Riyal (ريال سعودي), Omani Riyal (ريال عماني), Qatari Riyal (ريال قطري), Yemeni Riyal (ريال يمني) and Real Madrid (ريال مدريد), and based on the context of the document most related concepts are selected.

Table 3: Sample of weighted features in documents d_1 and d_2 with BOW representation model

Features	d_1		d_2	
	TF	TFIDF	TF	TFIDF
Riyal (ريال)	2	0.2568	1	0.01098
Saudi (سعودي)	2	0.1892	0	0
Madrid (مدريد)	0	0	1	0.0200

Table 3 shows a sample of features with their corresponding weights in both sample documents d_1 and d_2 with the BOW model. This example demonstrates how the word Riyal (ريال) and Madrid (مدرید) have low weights in document d_2 despite their importance in the classification step. Table 4 shows that Real Madrid (ريال مدريد) weight is increased in BOC model and reflect the importance of the concept in document d_2 .

Table 4: Sample of weighted features in two documents in BOC representation model

Features	d_1		d_2	
	TF	TFIDF	TF	TFIDF
Saudi Riyal (ريال سعودي)	2	0.4374	0	0
Real Madrid (ريال مدريد)	0	0	1	0.1359

3.3 Classification algorithms

In the last twenty years a wide range of ML algorithms have been used for TC. The ML algorithms automatically learn from a set of pre-classified (labelled) documents.

In this work two classification algorithms have been used, namely NB and Random Forest.

a. Naive Bayes Classifier: Naive Bayes (NB) is a probabilistic classifier based on applying Baye's theorem, and is commonly used in ML applications (Mitchell, 1996). The basic idea in a NB-based classifier is to estimate the probabilities of categories for a given document by observing the joint probabilities of features and categories.

b. Random Forest classifier: Random Forest is a commonly used classification method and it proposed by Breiman (2001). Random Forest builds a set of classification trees based on a subspace of features randomly selected to predict a category of a text instance.

4 EVALUATION

To evaluate the performance of the BOC representation model, we have assessed the accuracy of the developed classification system using two Arabic text classification datasets:

a. Arabic 1445 dataset: This dataset has been collected by Mesleh (2007) from online Arabic

newspaper archives including Al-Jazeera, Al-Nahar, Al-hayat, Al-Ahram, and Al-Dostor. The dataset contains 1445 documents that vary in length and fall into nine categories: Computer, Economics, Education, Engineering, Law, Medicine, Politics, Religion, and Sports.

b. Saudi Newspapers (SNP) dataset: The dataset consists of 5121 Arabic documents of different lengths which belong to seven categories: Culture, Economics, General, Information Technology, Politics, Social, and Sport. It has been collected by Al-Harbi et al. (2008) and consists of articles and news stories from Saudi newspapers.

We conducted all the experiments using WEKA (Hall et al., 2009), which is a popular open source toolkit for machine learning. We first converted the textual documents into the format required by WEKA, i.e., ARFF format (Attribute-Relation File Format)⁵. We then used this data to train two separate classification algorithms namely, NB and Random Forest. This was then followed by a ten-fold cross validation to test and evaluate the performance of the classifiers, in terms of Accuracy and Macro F1 measures. These performance measures are built upon the concepts of Precision (Pr) and Recall (Re). Precision is the probability that a document predicted to be in category c_i , truly belongs to this category. Recall is the probability that a document belonging to c_i is classified into this category. When a single performance measure is desired, the harmonic mean of the precision and recall, F1, is quoted. The Accuracy (Acc) is computed by dividing the total number of documents assigned to a given category c_i by the total number of documents in the testing dataset. Let $C = (c_1, \dots, c_i)$ denote the set of categories in the dataset, accordingly with respect to a given category c_i :

$$\begin{aligned} \text{Re}(c_i) &= \frac{\text{Number of correctly assigned class labels}}{\text{Total possible correct}} \\ &= \frac{TP_i}{TP_i + FN_i} \end{aligned} \quad (4)$$

$$\begin{aligned} \text{Pr}(c_i) &= \frac{\text{Number of correctly assigned class labels}}{\text{Total assigned}} \\ &= \frac{TP_i}{TP_i + FP_i} \end{aligned} \quad (5)$$

⁵ <http://www.cs.waikato.ac.nz/ml/weka/arff.html>

$$F1(c_i) = \frac{2Pr(c_i)Re(c_i)}{Pr(c_i) + Re(c_i)} \quad (6)$$

$$Acc(c_i) = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \quad (7)$$

where, the Re, Pr, F1 and Acc are computed in terms of the labels TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative), such that:

- TP_i : refers to the cases when both the classifier and human cataloguer agree on assigning category c_i to document d ;
- TN_i : refers to the cases when both the classifier and human cataloguer agree on not assigning category c_i to document d ;
- FP_i : refer to the cases when the classifier has mistakenly (as judged by a human cataloguer) has assigned category c_i to document d ;
- FN_i : refers to the cases when the classifier has failed (as judged by a human cataloguer) to assign a correct category c_i to document d .

For automatic text classification, precision and recall values for the various classes should be combined to obtain an accurate measure of the classifier algorithms used. This is commonly done using the Macro-averaged performance, which is calculated by first computing the scores per category, i.e., $Re(c_i)$, $Pr(c_i)$, $F1(c_i)$, and then averaging these per-category scores to compute the global means.

Figure 2 compares the NB-based classifier accuracy achieved by the BOW model and BOC model as applied to both Arabic datasets. Figure 3 on the other hand compares the same for the Random Forest based classifier.

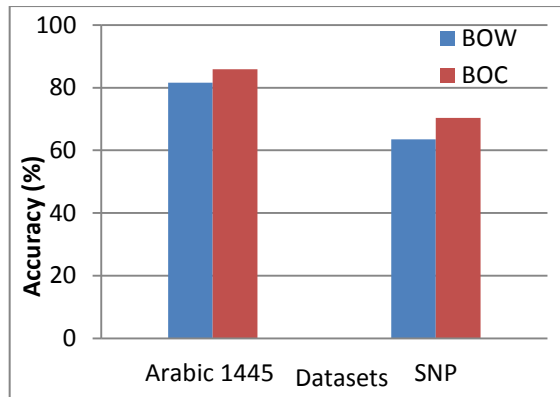


Figure 2: Ten-fold average accuracy achieved by the NB-based classifier for the BOW and BOC text representation models.

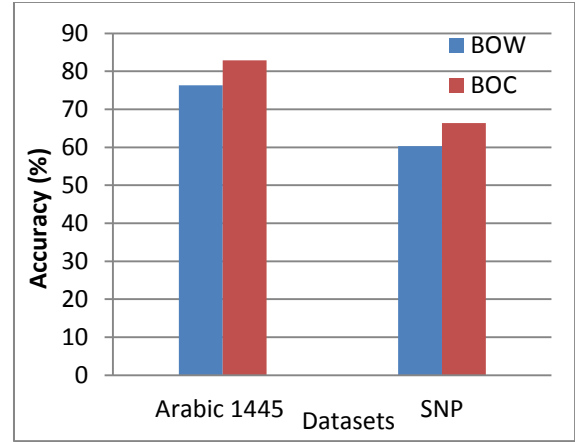


Figure 3: Ten-fold average accuracy achieved by the Random Forest -based classifier for the BOW and BOC text representation models.

Both Figures 2 and 3 indicate that the BOC representation used in our experiments for Arabic TC outperforms the BOW representation. The NB classifier seems to yield a 5% higher accuracy compared to the Random Forest classifier when applied to the Arabic 1445 dataset, and a 3% higher accuracy in the case of the SNP dataset. It has also been noted that the BOC based classifiers offer faster the execution time compared to that of BOW's. This is due to the fact that the BOC model tends to represent a document using fewer features compared to the BOW model. Table 5 gives details of the number of features used in each text representation model used in the NB classifier in our experiments and corresponding execution times of the classification algorithm in each case. In this experiment a desktop PC with a 3.33GHz Pentium processor operating under Windows 7 was used.

Table 5: Number of features and execution times for the NB-based classifier for both text representation models.

Model	Number of features	Classifier execution time (s)
BOW	12821	5.61
BOC	3463	0.96

Figure 4 shows the performance of the NB-based classifier, in terms of the Macro F1 measure, for both BOW and BOC models. The figure clearly shows that the BOC model outperforms the BOW by 10% when applied to the SNP dataset. In contrast to other classifiers that use the BOW

model, our NB-based classifier with the BOC model achieved 88.99% in Macro F1 measure when applied to the Arabic 1445 dataset, as compared to 84.53% achieved by NB-based classifier reported by Kanaan et al. (2009), and 88.11% achieved by the SVM-based classifier reported by Mesleh (2007). For the SNP dataset, our NB-based classifier utilising the BOC model achieved 80.24% in Macro F1 measure compared to 74.0% achieved by the same type of classifier reported by Alsaleem (2011).

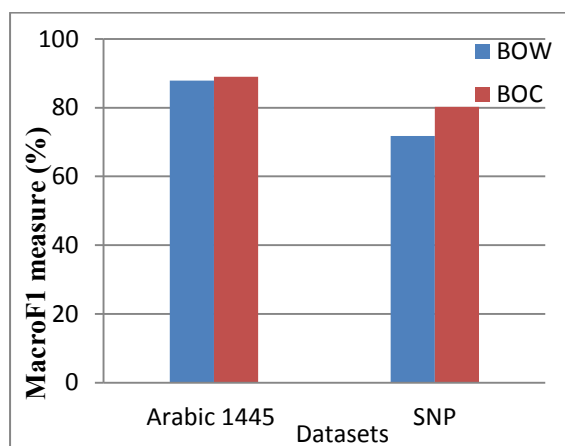


Figure 4: Performance of the NB-based classifier in terms of the Macro F1 (ten-fold averaged) for both the BOW and BOC text representation models.

5 CONCLUSIONS

In this work, we demonstrated that Arabic text classification can be improved by representing textual documents as a set of concepts using the BOC model. By doing so, background knowledge can be introduced to the document representation from sources such as Wikipedia to overcome some of the limitations of the classic BOW representation. As demonstrated in our reported experimental results, the described BOC text representation model significantly improves the classification accuracy compared to the BOW model when evaluated using two Arabic TC datasets.

REFERENCES

- AL-HARBI, S., ALMUHAREB, A., AL-THUBAITY, A., KHORSHEED, M. & AL-RAJEH, A. 2008. Automatic Arabic text classification.
- ALAHMADI, A., JOORABCHI, A. & MAHDI, A. E. A new text representation scheme combining Bag-of-Words and Bag-of-Concepts approaches for automatic text classification. GCC Conference and Exhibition (GCC), 2013 7th IEEE, 2013. IEEE, 108-113.
- ALAHMADI, A., JOORABCHI, A. & MAHDI, A. E. 2014. Combining Bag-of-Words and Bag-of-Concepts Representations for Arabic Text Classification. *IET Irish Signals & Systems Conference 2014*.
- ALSALEEM, S. 2011. Automated Arabic Text Categorization Using SVM and NB. *Int. Arab J. e-Technol.*, 2, 124-128.
- BLACK, W., ELKATEB, S. & VOSSSEN, P. Introducing the Arabic wordnet project. In Proceedings of the third International WordNet Conference (GWC-06, 2006. Citeseer.
- BREIMAN, L. 2001. Random forests. *Machine learning*, 45, 5-32.
- ELBERRICHI, Z. & ABIDI, K. 2012. Arabic Text Categorization: a Comparative Study of Different Representation Modes. *International Arab Journal of Information Technology (IAJIT)*, 9.
- GABRILOVICH, E. & MARKOVITCH, S. Feature generation for text categorization using world knowledge. *IJCAI*, 2005. 1048-1053.
- GABRILOVICH, E. & MARKOVITCH, S. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. *AAAI*, 2006. 1301-1306.
- HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P. & WITTEN, I. H. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11, 10-18.
- HOTH, A., STAAB, S. & STUMME, G. 2003. Wordnet improves Text Document Clustering.
- ISO. (2009). ISO-704: Terminology work—Principles and methods (3rd ed.). Geneva, Switzerland: International Organization for Standardization.
- KANAAN, G., AL-SHALABI, R., GHWANMEH, S. & AL-MA'ADEED, H. 2009. A comparison of text-classification techniques applied to Arabic text. *Journal of the American society for information science and technology*, 60, 1836-1844.
- MESLEH, A. M. D. A. 2007. Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System. *Journal of Computer Science*, 3.
- MILNE, D. & WITTEN, I. H. 2013. An open-source toolkit for mining Wikipedia. *Artificial Intelligence*, 194, 222-239.
- MITCHELL, T. 1996. Machine Learning. *McCraw Hill*.

- SALTON, G. & BUCKLEY, C. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24, 513-523.
- VERSTEEGH, K. & VERSTEEGH, C. 1997. *The Arabic Language*, Columbia University Press.

Combining Bag-of-Words and Bag-of-Concepts Representations for Arabic Text Classification

Alaa Alahmadi¹, Arash Joorabchi², Abdulhussain E. Mahdi³

Department of Electronic & Computer Engineering

University of Limerick, Limerick, Ireland

Email: {¹alaa.alahmadi; ²arash.joorabchi; ³hussain.mahdi}@ul.ie

Abstract— This paper introduces a set of new approaches for text representation for automatic classification of Arabic textual documents. These approaches are based on combining the well-known Bag-of-Words (BOW) and the Bag-of-Concepts (BOC) text representation schemes and utilizing Wikipedia as a knowledge base. The proposed representations are used to generate a vector space model, which in turn is fed into a classifier to categorize a collection of Arabic textual documents. Three different machine learning based classifiers have been utilized in this work. Performance of proposed text representation models is evaluated in comparison to using a standard BOW scheme and a concept-based scheme, as well as recently reported similar text representation schemes that are based on augmenting the standard BOW with the BOC.

Keywords – Arabic Text Classification; Natural Language Processing; Wikipedia.

I INTRODUCTION

Text classification (TC) is considered one of the important subfields of data mining as it plays an important role in information extraction and summarization, text retrieval, and question answering. TC is the task of assigning one or more predefined categories to a given textual document. The classification performance and accuracy is affected by the text representation model utilised in the process. The traditional representational scheme in TC is the Bag-of-Words (BOW) [1]. The approach represents text as a vector $\vec{d}$ of word weights, where each document, $d_i \in D$, is expressed as a weighted high dimensional vector, $\vec{d}_i$, and each dimension corresponds to a unique word. However, when applied to a document, this representation tends to divide text into single words and, hence, causing terms to break down into their constituent words. For example, it breaks the term ‘mechanical engineering’ (هندسة ميكانيكية) into the words ‘engineering’ (هندسة) and ‘mechanical’ (ميكانيكية). The model also treats synonymous words as independent features, e.g. ‘record’ (وثيقة) and ‘document’ (مستند) are considered as two independent words with no semantic association. These issues have been addressed by a number of researchers by representing text as concepts rather than words, using an approach

known as Bag-of-Concepts (BOC) [2-4]. In the BOC approach, semantic knowledge bases such as WordNet¹, Open Directory Project (ODP)², and Wikipedia³ are used to identify the concepts appearing within a document.

Wikipedia is the largest electronic knowledge repository currently available on the Internet. It contains over 4 million articles which are contributed collaboratively by volunteers. Each article in Wikipedia describes a single topic and the article title represents a concept term. The great majority of Wikipedia concepts are classified according to Wikipedia classification system, which is hierarchical. All these features make Wikipedia a rich source to identify concepts to build a BOC model for a classification system and is a potential ontology for enriching text representation. The BOC approach captures and preserves the semantics and associations between words appearing in the document. On the other hand, it has been shown that the concepts alone do not provide a sufficient representation of text as they do not cover all words in a document [3, 4].

In previous work, we developed a number of new text representation models, which combine the BOW

¹ <http://wordnet.princeton.edu/>.

² <http://dmoz.org>.

³ <http://www.wikipedia.org>.

and the BOC schemes, and applied them to classification of English text [5]. This current work focuses on applying these text representation models to Arabic textual documents, using Arabic Wikipedia as a thesaurus. To the best of our knowledge, we are the first to use Arabic Wikipedia as a semantic knowledge base to enrich and overcome some of the limitations of the traditional BOW model for Arabic TC. The remainder of the paper is organized as follows: Section II reviews related work in Arabic TC. Section III describes the BOW and BOC text representational schemes and illustrates their use. Section IV describes existing approaches for enriching the BOW model for English text and then introduces our new combined models for text representation. Section V then describes the experimental set-up for evaluating the performance of our proposed combined models in automatic Arabic TC and compares that to the schemes presented in Section IV. The paper concludes in Section VI.

II RELATED WORK

Arabic belongs to the Semitic group of languages, and it is a highly inflectional and derivational language [6]. The Arabic alphabet consists of 28 characters. These characters are directly connected when forming words and there exist no capital letters, unlike western scripts. Unlike English, Arabic text is read from right to left.

There are a number of studies which apply the BOW approach with different method to Arabic TC. For example; the author in [7] compared three classification algorithms which use the BOW model with a Chi Square statistics for feature selection. Their best algorithm, which uses a Support Vector Machine (SVM) classifier, yields 88.11%, in Macro F1 measure when applied to an in-house compiled Arabic documents set. Other researchers [8] have compared the performance of k-Nearest Neighbors (kNN), Rocchio, and Naïve Bayes (NB) classifiers for TC of Arabic documents set compiled by [7]. In their work, a different weighting scheme is applied to find the most efficient combination, such as Term Frequency (TF), Term Frequency Inverse Document Frequency (TFIDF), Inverse Category Frequency (ICF), and Weighted IDF (WIDF). Their reported results show the NB classifier to be superior to the other two classifiers, yielding 84.53% in Macro F1 measure. As for weighting methods, the *WIDF* scheme provides the best performance on kNN, while *TFIDF* shows the best performance on Rocchio. Other applies the BOC approach on Arabic TC such as [10] where a comparison between different text representations models such as BOW, N-grams and BOC is performed. Arabic WordNet (AWN) [11] is used as a lexical and semantic resource to build BOC and kNN classification algorithms as a classifier. Other used topic modelling

approach such as in [12] Latent Dirichlet Allocation (LDA) was used to represent documents as random mixtures over latent topics, where each topic is characterized by a distribution over words.

III DOCUMENT REPRESENTATION

Prior to applying TC, a textual document needs to be represented using features extracted from the labelled documents (Training Set). These features could be just words and in this case, a BOW representation is created for the document. On the other hand, the features could be a set of concepts creating a BOC representation for the same document. In order to use Wikipedia concepts to build a BOC representation, a toolkit known as Wikipedia-Miner⁴ [15] may be used. This toolkit processes Wikipedia dump files and extracts all the concepts and their semantic relations. It allows comparing words and concepts semantically and detecting Wikipedia concepts within documents. The Wikipedia-Miner has already been applied to a wide range of research problems in the area of text mining.

When creating a text representation model, each feature need to be weighted using the Term Frequency–Inverse Document Frequency (*TFIDF*) [14]:

$$TFIDF(f_m, d_i) = TF(f_m, d_i) \cdot IDF(f_m) \quad (1)$$

where $TF(f_m, d_i)$ is a frequency of a feature, $f_m \in F$, defined as:

$$TF(f_m, d_i) = \frac{n(f_m, d_i)}{|d_i|} \quad (2)$$

with $n(f_m, d_i)$ being the occurrence frequency of the feature f_m in document d_i , normalized by $|d_i|$, which is the length of d_i . The Inverse Document Frequency, $IDF(f_m)$, is defined as:

$$IDF(f_m) = \log_e \frac{|D|}{DF(f_m)} \quad (3)$$

where $DF(f_m)$ is the document frequency of the feature f_m . It counts the number of documents in D where the feature f_m appears. $|D|$ is the total number of documents in the corpus. The $IDF(f_m)$ parameter has the effect of reducing the weight of those words which appear in a large portion of the corpus D .

For example, consider a document d_1 , which discusses the subject of ‘information retrieval’ (استرجاع المعلومات) and a second document d_2 , which discusses the subject of ‘search engines’ (محرك بحث) and both documents d_1 and d_2 belonging to the same class. Table 1 show a sample of the traditional BOW representation model for these two documents, where feature values correspond to term frequencies.

⁴ Available at <http://wikipedia-miner.cms.waikato.ac.nz/>

Table 1: BOW approach

Word	d_1	d_2
(search) بحث	0	2
(information) المعلومات	2	0
(retrieval) استرجاع	1	0
(computer) الحاسب	1	1
(data) بيانات	1	1
(record) وثيقه	0	2
(document) مستند	3	0

Using feature values corresponding to term frequencies again, Table 2 shows a sample of the BOC representation model for the same documents, where the features are candidate concepts identified in document. Inspection of Tables 1 & 2 shows that the term 'information' (المعلومات), for example, appears in the BOW model twice, whereas it appears once in BOC model, as indicated by the term frequencies for 'information retrieval' and 'information science' in Table 2.

Table 2: BOC approach

Concepts	d_1	d_2
(information retrieval) استرجاع المعلومات	1	0
(search engines) محرك بحث	0	2
(document) مستند	3	2
(computer science) علم الحاسب	1	1
(information science) علم المعلومات	1	2

It is clear that the BOC representation reduces the term frequency of features as it keeps multi words term. It also has less features compared to BOW representation due to the fact that concepts do not cover all words in a given document.

IV THE COMBINED MODEL

By enriching the BOW model with concepts from the Wikipedia using different strategies via a combined model has three benefits: (1) it resolves synonyms; (2) it introduces more general concepts which help to identify related topics; and (3) it resolves the low performance of BOC when used alone. In this work, we have investigated the realisation of the BOW-BOC combined model using the following approaches; using three of the models which have been designed previously in [3, 4] for English TC and two new models which have been created exclusively for our previous work in English [5].

a) *Adding Concepts (AC)*: this approach involves adding the concepts identified in the document to its BOW model as extra features and was first proposed by the Hotho et. al. [3]. This approach has since been used by a number of other researchers who demonstrated its improving impact on the performance of standard English text classifiers [2, 4, 16].

Table 3: AC approach

Entity	d_1	d_2
(search) بحث	0	2
(information) المعلومات	2	0
(retrieval) استرجاع	1	0
(computer) الحاسب	1	1
(data) بيانات	1	1
(record) وثيقه	0	2
(document) مستند	3	0
(information retrieval) استرجاع المعلومات	1	0
(search engines) محرك بحث	0	2
(computer science) علم الحاسب	1	1
(information science) علم المعلومات	1	2

b) *Replacing Terms with Concepts (RTC)*: this approach is similar to AC, but with the following addition. Words in the document for which a Wikipedia concept has been identified are removed from the combined text representation model. The approach was first proposed by [3] for English TC and used in [10] for Arabic TC with AWN for concepts.

Table 4: RTC approach

Entity	d_1	d_2
(data) بيانات	1	1
(information retrieval) استرجاع المعلومات	1	0
(search engines) محرك بحث	0	2
(computer science) علم الحاسب	1	1
(information science) علم المعلومات	1	2

Table 4 shows all concepts from Table 2 and only words from Table 1 which do not have corresponding concepts. The main difference between the AC and the RTC is that when using the RTC approach, the combined vector will contain more concepts than words.

c) *Adding Concepts and Categories (ACC)*: in this approach, the parent categories of concepts are added to the combined feature vector of the representation model, along with concepts and words. This approach was first proposed in [4] for English TC.

d) *Adding Unmapped Concepts (AUC)*: this approach simply involves taking the basic BOW and adding to it the concepts whose representative words do not explicitly appear in the BOW model. As we can see in Table 5, all words from Table 1 are added to the combined vector of the representation model, but only 'information science' (علم المعلومات) concept from the set of concepts shown in Table 2 is added.

Table 5: AUC approach

Word		d_1	d_2
(search)	بحث	0	2
(information)	المعلومات	2	0
(retrieval)	استرجاع	1	0
(computer)	الحاسب	1	1
(data)	بيانات	1	1
(record)	وثيقه	0	2
(document)	مستند	3	0
(information science)	علم المعلومات	1	2

The concept ‘information science’ (علم المعلومات) has been detected in document as related to both concepts ‘information retrieval’ (استرجاع المعلومات) and ‘search engines’ (محرك بحث) concepts. Hence, it is added to the BOW feature vector despite it not appearing in the text of the two documents. This way, the BOW vector is enriched with the general concepts that will be added using this approach.

To be able to use this approach a dictionary needs to be built after extracting features from Training Set. Figure 1 outline the process of creating the AUC dictionary.

```

Algorithm to build AUC dictionary (concepts-dictionary  $C$ ;
words-dictionary  $W$ ; concept-terms Map  $CTM$ )
Create empty AUC dictionary ( $AUC-D$ )
Add all features in  $W$  to RUC-d
For each concept,  $c_j \in C$  do
    If ( $c_j \notin AUC-D$ ) and ( $c_j \in CTM$ )
        Get  $c_j$  frequency from  $C$  and add  $c_j$  to RUC-d
    End if
    Else if ( $c_j \in AUC-D$ ) and ( $c_j \in CTM$ )
        Get  $c_j$  frequency from  $C$  and update  $c_j$  frequency
    in AUC-d
    End if
End for
Return AUC dictionary ( $AUC-D$ )

```

Figure 1: Create AUC dictionary

First, all concepts and words are extracted as features from the Training Set. Then, each concept is mapped to its corresponding word/s. By doing this, the algorithm will resolve synonyms and capture different words which refer to the same concept in documents of the Training Set. Next, all words that are considered as features for the BOW model are added to the dictionary. In this way, words which have significant frequency value and are not represented by concepts are kept in the AUC model. Also, each concept that does not appear as a word in the BOW model and not mapped directly to a word(s) is added to the AUC dictionary.

- e) *Using Concepts for Terms which do not appear in the Document (CTD):* in a similar way to the

BOW scheme, this approach works by using words for representing a document. The only difference is in the weighting scheme used, where all the words from the dictionary, which do not appear in the document but have a corresponding concept in the documents BOC vector, would be added to the combined vector with weights equal to those of the corresponding concepts.

```

Algorithm to weight features in CTD document vector (document
word vector  $d_w$ ; document concepts vector  $d_c$ ; BOW dictionary
 $BOW-D$ , term-concepts Map  $TCM$ )
For each feature,  $f_j \in BOW-D$  do
    If ( $f_j \in d_w$ )
        Get  $f_j$  weight in  $d_w$  and used it in corresponds
        dimensional for the new
        document vector
    End if
    Else
        If ( $f_j \in TCM$ )
            Get  $f_j$  corresponding concept  $f_c$ 
            if ( $f_c \in d_c$ )
                Get  $f_c$  weight in  $d_c$  and used it in corresponds
                dimensional for new
                document vector
            End if
        End if
    End if
End for
Return weighted features CTD document vector

```

Figure 2: weighting scheme for CTD Approach

Figure 2 illustrates the weighting scheme. First, a ‘Word-Concept’ map or dictionary needs to be created. All concepts from the training documents are counted and mapped to their corresponding words from the documents. In most cases a concept is mapped to one word. However, depending on the concept, if it is a multi-word or it has synonyms, it will be mapped to two or more words. To represent a document by CTD approach. First, each word from the BOW dictionary should be checked. If it appears in the document, the weight of that word is set using equation (1). Otherwise, the corresponding concept for the word is retrieved from the Word-Concept map and the documents BOC vector will be searched for that concept. If a concept exists, the word will be assigned the weight of the concept. If it does not exist, the word will not be represented in the combined vector.

Table 6: CTD approach

Word		d_1	d_2
(search)	بحث	0	2
(information)	المعلومات	2	0
(retrieval)	استرجاع	1	0
(computer)	الحاسب	1	1
(data)	بيانات	1	1
(record)	وثيقه	3	2
(document)	مستند	3	2

Table 6, shows both documents d_1 and d_2 represented by the CTD approach. All words

V EVALUATION

To represent the documents using the different text representation approaches described in section IV, the documents first need to be pre-processed using the following steps:

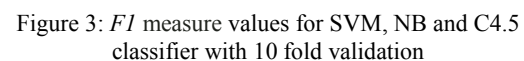
- In these experiments, we used the linear SVM, NB and C4.5 decision tree algorithms to train the model to classify the documents. We conducted all the experiments using the software WEKA [17], which is a popular open source toolkit that provides an implementation for different classification algorithms. We evaluated performance of the proposed approaches using the Micro and Macro *F1*

Given a class c_i , *precision* (c_i) is defined as:

and *recall* (c_i) defined as:

The *FI* measure is the harmonic mean of the two:

The micro-averaged $F1$ measure operates at the document level, and is primarily affected by the categorization accuracy of the larger categories. On the other hand, the macro-averaged $F1$ measure averages results over all categories.



IPR2025-01398

In addition, representing documents using concepts rather than words with the NB as classification algorithm seems to provide a better performance in conjunction with the BOC model compared to BOW model.

By using Chi-Square feature selection method [18] to reduce number of features (dimensions) in models, the *F*-measure for the SVM-based classifier drops in all models except for AC and CTD combined models which accomplished best performance. This is illustrated in the chart shown in Figure 4.

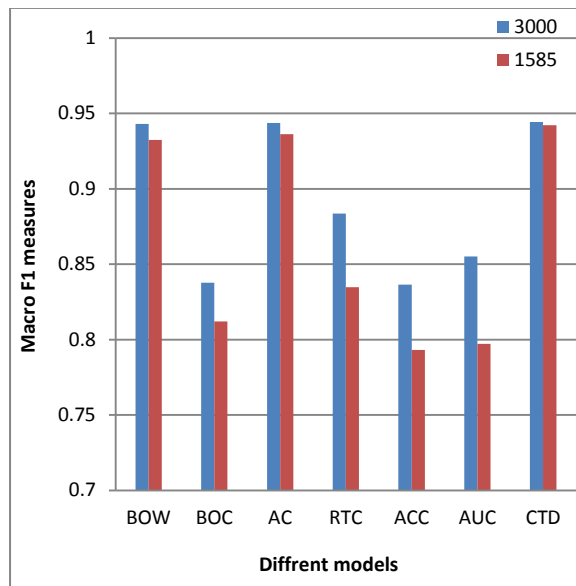


Figure 4: *F1* measure values for SVM classifier with CHI method at different sizes of features

VI CONCLUSION

In this work, we demonstrated that Arabic text classification can be improved by enriching the representation models of these documents with related concepts. Accordingly, we presented and proposed a number of different representation approaches by combining the BOW and the BOC text representation schemes.

To the best of our knowledge, we are the first to use Wikipedia as knowledge base to overcome some of the limitations of the classic BOW representation in Arabic TC. We have described five different models for text representation which combine and enrich the BOW with BOC models and experimented with different classification algorithms, from which SVM was the best performer, followed by NB and C4.5, respectively. Presented experimental results demonstrated superior performance by all five models when applied to Arabic TC, as compared to the BOW alone. Among these, the AC and CTD Models, have been shown to achieve higher categorization accuracy compared to others. In addition, our CTD model achieved a stable performance even when reduce the number of features. Finally, it should be noted here that, to the

best of our knowledge, our current work represents the first attempt at applying the AC and ACC approaches to Arabic TC using Arabic Wikipedia.

REFERENCES

- [1] Z. S. Harris, "Distributional structure," in Papers in structural and transformational linguistics, ed: Springer, 1970, pp. 775-794.
- [2] E. Gabrilovich and S. Markovitch, "Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge," in AAAI, 2006, pp. 1301-1306.
- [3] A. Hotho, S. Staab, and G. Stumme, "Wordnet improves Text Document Clustering," 2003.
- [4] P. Wang, J. Hu, H.-J. Zeng, and Z. Chen, "Using Wikipedia knowledge to improve text classification," Knowledge and Information Systems, vol. 19, pp. 265-281, 2008.
- [5] A. Alahmadi, A. Joorabchi, and A. E. Mahdi, "A new text representation scheme combining Bag-of-Words and Bag-of-Concepts approaches for automatic text classification," in GCC Conference and Exhibition (GCC), 2013 7th IEEE, 2013, pp. 108-113.
- [6] K. Versteegh and C. Versteegh, The Arabic Language: Columbia University Press, 1997.
- [7] A. M. d. A. Mesleh, "Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System," Journal of Computer Science, vol. 3, 2007.
- [8] G. Kanaan, R. Al-Shalabi, S. Ghwanmeh, and H. Al-Ma'adeed, "A comparison of text-classification techniques applied to Arabic text," Journal of the American society for information science and technology, vol. 60, pp. 1836-1844, 2009.
- [9] L. Khreisat, "A machine learning approach for Arabic text classification using N-gram frequency statistics," Journal of Informetrics, vol. 3, pp. 72-77, 2009.
- [10] Z. Elberichi and K. Abidi, "Arabic Text Categorization: a Comparative Study of Different Representation Modes," International Arab Journal of Information Technology (IAJIT), vol. 9, 2012.
- [11] W. Black, S. Elkateb, and P. Vossen, "Introducing the Arabic wordnet project," in In Proceedings of the third International WordNet Conference (GWC-06, 2006.
- [12] M. Zrigui, R. Ayadi, M. Mars, and M. Maraoui, "Arabic Text Classification Framework Based on Latent Dirichlet Allocation," Journal of Computing and Information Technology, vol. 20, pp. 125-140, 2012.
- [13] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," Communications of the ACM, vol. 18, pp. 613-620, 1975.

- [14] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, pp. 513-523, 1988.
- [15] D. Milne and I. H. Witten, "An open-source toolkit for mining Wikipedia," *Artificial Intelligence*, vol. 194, pp. 222-239, 2013.
- [16] L. Jing, M. K. Ng, and J. Z. Huang, "Knowledge-based vector space model for text clustering," *Knowledge and Information Systems*, vol. 25, pp. 35-55, 2009.
- [17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, pp. 10-18, 2009.
- [18] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, pp. 1-47, 2002.

A New Text Representation Scheme Combining Bag-of-Words and Bag-of-Concepts Approaches for Automatic Text Classification

Alaa Alahmadi, Arash Joorabchi and Abdulhussain E. Mahdi

Electronic and Computer Engineering Department, University of Limerick, Ireland
 {alaa.alahmadi; arash.joorabchi; hussain.mahdi}@ul.ie

Abstract— This paper introduces a new approach to creating text representations and apply it to a standard text classification collections. The approach is based on supplementing the well-known Bag-of-Words (BOW) representational scheme with a concept-based representation that utilises Wikipedia as a knowledge base. The proposed representations are used to generate a Vector Space Model, which in turn is fed into a Support Vector Machine classifier to categorise a collection of textual documents from two publically available datasets. Experimental results for evaluating the performance of our model in comparison to using a standard BOW scheme and a concept-based scheme, as well as recently reported similar text representations that are based on augmenting the standard BOW approach with concept-based representations.

Keywords— *Text Classification; Bag-of-Words; Bag- of-Concepts; Wikipedia.*

I. INTRODUCTION

Text or document classification is the task of assigning one or more predefined categories to a given textual document via applying appropriate natural language processing techniques. More formally, if d_i is a document from a given set of documents D , and $C = \{c_1, c_2, \dots, c_n\}$ is the set of available categories, then the task of the classifier is to assign one or more categories, c_j , to the document d_i . A crucial element that decides the performance and accuracy of the categorisation is the text representational model utilised in the process. The most common representational scheme in text categorization is the Bag-of-Words (BOW) approach, in which a text is represented as a vector $\vec{d}$ of words weights [1]. However, this representation suffers from two main limitations: (1) it breaks terms into their constituent words, e.g. it breaks ‘Text Classification’ into the words ‘Text’ and ‘Classification’; (2) it treats synonymous words as independent features, e.g. ‘Classification’ and ‘Categorization’ are considered as two independent words with no semantic association. Researchers have attempted to address above issues by representing text as concepts rather than words, using an approach known as Bag-

of-Concepts (BOC) [5-8]. The main benefit of the BOC approach is that it captures and preserves the semantics and associations between words appearing in the document. In this approach, semantic knowledge bases such as WordNet [2], Open Directory Project (ODP)¹ and Wikipedia² are used to identify the concepts appearing within a document.

Authors in [4] utilized synonyms from WordNet of each term as the related concepts. The results of this method show efficiency in some cases but not in all cases as WordNet is limited since it is a manually constructed dictionary and therefore laborious to maintain. To deal with this problem, other researches have explored the usage of a knowledge base derived from the Internet, such as ODP and Wikipedia, for developing various feature generation and text processing techniques, e.g., see [5, 6]. The techniques reported in [5, 6] have shown improved performance when applied to text classification. Their results have also shown that Wikipedia is less noisy than ODP when used as a knowledge base. Yet, the approach ignores the structural relations within Wikipedia, such as hyperlinks and hierarchical categories. Authors in [7, 8] have addressed this issue by constructing an informative thesaurus from Wikipedia, so that the synonymy, polysemy, hypernymy, and associative relations between concepts can be clearly derived.

Wikipedia is the largest electronic knowledge repository on the Internet containing millions of articles contributed collaboratively by volunteers [3]. Wikipedia is a comprehensive, up to date, and is well-formed knowledge source; however, it is not a structured thesaurus similar to WordNet or MeSH³. Each article in Wikipedia describes a single topic and the article title represents a concept term. The great majority of Wikipedia concepts are classified according to Wikipedia classification system, which is hierarchical and includes a total of N classes. For these reasons Wikipedia is a rich source to identify concepts to build a BOC vector for a classification system and is a potential ontology for enriching text representation.

¹ <http://wordnet.princeton.edu/>.

² <http://www.wikipedia.org>.

³ <http://www.nlm.nih.gov/mesh/>

On the other hand, it has been shown that the concepts alone do not provide a sufficient representation of text as it does not cover all words in a document [4]. In this work we investigate the development of means for enriching the BOW text representation with concepts and categories, as derived from the Wikipedia as a knowledge base. Specifically, we propose a new text representation scheme which combines the BOW and the BOC schemes, using Wikipedia as a thesaurus.

The remainder of the paper is organized as follows: Section II describes the BOW and BOC text representational schemes and illustrates their use. Section III describes existing approaches for enriching the BOW model and then introduces our new combined model for text representation. Section IV then describes the experimental set-ups for the evaluating the performance of proposed combined model in automatic text classification and compare that to the schemes presented in Section III. The paper concludes in Section V.

II. DOCUMENT REPRESENTATION

Documents are traditionally represented using a vector space model [14]. Let D be a set of documents, and F the set of features representing these documents. Each document, $d_i \in D$, is expressed as a weighted high dimensional vector, $\vec{d}_i$, and each dimension corresponds to a unique feature, such that $\vec{d}_i = (f_1, f_2, \dots, f_m)$, where, $f_m \in F$, is the weight of a given feature of that document. In the case of the BOW, the features are the words, w_m , contained in the document. Let W be the set of all different words occurring in D and w_m is a word, where, $w_m \in W$. Typically, W is constructed after removing stop words, and after stemming terms to their roots. The Term Frequency-Inverse Document Frequency (TFIDF) [13], formula used to compute the weight of word w_m :

$$TFIDF(w_m, d_i) = TF(w_m, d_i) \times IDF(w_m) \quad (1)$$

where $TF(w_m, d_i)$ is a frequency of a word, $w_m \in W$, is defined as:

$$TF(w_m, d_i) = \frac{n(w_m, d_i)}{|d_i|} \quad (2)$$

where $n(w_m, d_i)$ is the occurrence frequency of the word w_m in document d_i , normalized by $|d_i|$, which is the length of d_i . The inverse document frequency $IDF(w_m)$ is defined as:

$$IDF(w_m) = \log_e \frac{|D|}{DF(w_m)} \quad (3)$$

where $DF(w_m)$ is the document frequency of the word w_m . It counts the number of documents in D where the word w_m appears. $|D|$ is the total number of documents in the corpus. The $IDF(w_m)$ parameter has the effect of reducing the weight of those words which appear in a large portion of the corpus D . Considering d_1 and d_2 as two sample documents from corpus D , figure 1 and 2 show snippets of these two documents. The first document discusses the subject of document classification, whereas the second document discusses search engines and both documents belong to the same class.

Document classification or document categorization is a problem in natural language processing, library science and information science. The task is to assign a document to one or more classes or categories. This may be done 'manually' (or 'intellectually') or algorithmically. The documents to be classified may be texts, images, music, etc. A classifier ask himself: 'Under which descriptors should this entity be found?' and 'think of all the possible queries and decide for which ones the entity at hand is relevant'

Figure 1. A paragraph from the first sample document (d_1)

A web search engine is a software system that is designed to search for information on the World Wide Web. The information may be a specialist in web pages, images, information and other types of files. Some search engines also mine data available in databases or open directories. Unlike web directories, which are maintained only by human editors, search engines also maintain real-time information by running an algorithm on a web crawler.

Figure 2. A paragraph from the second sample document (d_2)

Using BOW scheme, each document d_1 and d_2 will be first pre-processed by removing stop words and stemming all the words to their root. Then the occurrence frequency of each unique stemmed word inside the document will be counted. TABLE I, shows words from both documents d_1 and d_2 along with their frequency within their corresponding document.

TABLE I. BOW APPROACH

Word	d_1	d_2
document	5	1
categorization	4	0
natural	1	0
language	1	0
query	1	0
algorithm	2	1
search	0	6
engine	0	3

In order to use Wikipedia concepts to make a BOC, a toolkit known as WikipediaMiner⁴ [9] may be used. This toolkit processes Wikipedia dump files and extracts all the concepts and their semantic relations. It allows comparing words and concepts semantically and detecting Wikipedia concept within documents. It has already been applied to a wide range of research problems in the area of text mining. TABLE II, illustrates concepts that have been identified in the two sample documents d_1 and d_2 using the WikipediaMiner. The concepts will be kept without performing stemming or breaking multi-

⁴ Available at <http://wikipedia-miner.cms.waikato.ac.nz/>

word terms. The identified concepts are then weighted using the TFIDF formula given in Equation (1).

TABLE II. BOC APPROACH

Entity	d_1	d_2
search engines	1	6
document classification	4	0
natural language processing	1	0
information retrieval	3	5

III. THE COMBINED MODEL

Enriching the BOW vector with concepts from the Wikipedia using different strategies via a combined model has three benefits: (1) it resolves synonyms; (2) it introduces more general concepts which help to identify related topics; and (3) it resolves the low performance of BOC when used alone. In this work, we have investigated the realisation of the BOW-BOC combined model using the following approaches; using three of the models which have been designed previously in [4, 8] and two new models which have been created expressly for this study.

A) Adding Concepts (AC)

The AC approach consists of adding the concepts identified in the document to its BOW vector as extra features, and was first proposed by the authors in [4]. This approach has since been used by a number of other researchers who demonstrated its improving impact on the performance of standard text classifiers [7, 8].

TABLE III. Adding Concepts (AC) Approach

Entity	d_1	d_2
document	5	1
categorization	4	0
natural	1	0
language	1	0
query	1	0
algorithm	2	1
search	0	6
engine	0	3
Search engines	1	6
document classification	4	0
natural language processing	1	0
information retrieval	3	5

TABLE III, shows words and concepts occurring in documents d_1 and d_2 . As can be seen, the word feature vectors shown in TABLE I has been extended by sets of concepts in TABLE II, and the feature weights are kept unchanged.

B) Replacing Terms with Concepts (RTC)

This approach is similar to Adding Concepts (AC), but in this approach those words in the document for which a Wikipedia concept has been identified will be removed from the combined vector [4].

TABLE IV. Replacing Terms with Concepts (RTC)

Entity	d_1	d_2
algorithm	2	1
search engines	1	6
document classification	4	0
natural language processing	1	0
information retrieval	3	5

TABLE IV shows all concepts from TABLE II and only words from TABLE I which do not have corresponding concepts. The main difference between AC and RTC is that using the RTC approach the combined vector will contain more concepts than words.

C) Adding Concepts and Categories (ACC)

This approach is similar to the ‘Adding Concepts’ but instead of adding only the concepts to the BOW the parent categories of concepts will also be added to the combined feature vector [8].

D) Adding Unmapped Concepts (AUC)

This approach simply involves adding concepts which their representative words do not appear in the BOW vector explicitly. As we can see in TABLE V, all words from TABLE I are added to the combined vector but only ‘Information Retrieval’ from the set of concepts shown in TABLE II is added.

TABLE V. Adding Unmapped Concepts (AUC) Approach

Entity	d_1	d_2
document	5	1
categorization	4	0
natural	1	0
language	1	0
query	1	0
algorithm	2	1
search	0	6
engine	0	3
Information retrieval	3	5

The concept ‘Information Retrieval’ has been detected in both documents as both concepts ‘document classification’ and ‘Search engines’ are applications of Information retrieval even if it does not appear in the text. The BOW vector will gain from general concepts that will be added using this approach.

E) Using Concepts for Terms which do not Appear in the Document (CTD)

In a similar way to the BOW approach, this approach works by using words for representing a document. The only difference is in the weighting scheme used, where all the words from the dictionary, which do not appear in the document but have a corresponding concept in the documents BOC vector, would be added to the combined vector with weights equal to those of the corresponding concepts. Figure.3 illustrates the weighting scheme.

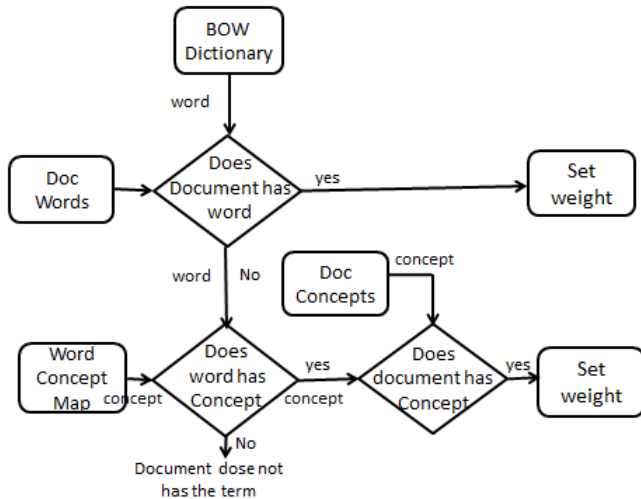


Figure.3. weighting scheme for CTD Approach

First, a ‘Word-Concept’ map dictionary needs to be created. All concepts from the training documents will be counted and mapped to their corresponding words from the documents. In most cases a concept will be mapped to one word. However, depending on the concept, if it is a multi-word or it has synonyms it will be mapped to two or more words.

To represent a document by CTD approach, first, each word from the BOW dictionary should be checked. If it appears in the document, word weight will be set using equation (1). Otherwise, the corresponding concept for the word will be retrieved from the Word-Concept map and the documents BOC vector will be searched for that concept. If a concept exists, the word will be assigned the weight of the concept. If it is not exist, the word will not be represented in the combined vector.

TABLE VI. Using Concepts for Terms which do not appear in the Document (CTD)

words	d ₁	d ₂
document	5	1
categorization	4	0
natural	1	0
language	1	0
query	1	0
algorithm	2	1
search	1	6
engine	1	3

TABLE VI. shows both documents d_1 and d_2 represented by the CTD approach. All words from TABLE I appear in the representation but the weight has been adjusted using concept weight from TABLE II. In d_1 both terms ‘search’ and ‘engines’ have been assigned a zero weight but in the new scheme they have been assigned the weight of their corresponding concept ‘Search Engines’.

IV. EXPERIMENTS

We have used two standard text classification datasets to evaluate the performance of our approach for improving document representation in text classification. In the following, a description of each dataset is provided:

1. 20 Newsgroups⁵ (20NG) [10]. The 20 Newsgroups-118828 data set is a popular collection of approximately 20,000 newsgroup documents, partitioned nearly evenly across 20 different newsgroups (about 1,000 documents per class). We split the data set into 80% training data and 20% testing data per class.
2. Classic 3⁶ dataset [12]. The documents consist of titles and abstracts from academic papers from three different subjects. All documents are retained, resulting in about 4000 documents.

In these experiments we used the linear Support Vector Machine (SVM) algorithm to train the model to classify the documents into their corresponding category. We conducted all the experiments using the software LIBSVM [11], and a linear kernel. We evaluated performance of the proposed approaches using Micro and Macro F1 measures for 20NG and Classic 3. When using the classifiers to predict the class of a test instance, there are four possible outcomes; True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Positive means that the document was classified as belonging to the category, negative that it was not, whereas true means that the classification was correct and false that it was not. From these four outcomes, we can define the standard evaluation metrics *precision* and *recall*.

Let $C = (c_1, \dots, c_i)$ denote the set of classes in the data set. Given a class c_i , *precision* (c_i) and *recall* (c_i) are defined as:

$$p(c_i) = \frac{TP}{TP+FP} \quad (4)$$

$$R(c_i) = \frac{TP}{TP+FN} \quad (5)$$

⁵ Available at <http://people.csail.mit.edu/jrennie/20Newsgroups/>

⁶ Available at <http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/>

The $F1$ measure is the harmonic mean of the two:

$$F1 = \frac{2P(C_i)R(C_i)}{P(C_i)+R(C_i)} \quad (6)$$

The micro-averaged $F1$ measure operates at the document level, and is primarily affected by the categorization accuracy of the larger categories. On the other hand, the macro-averaged $F1$ measure averages results over all categories.

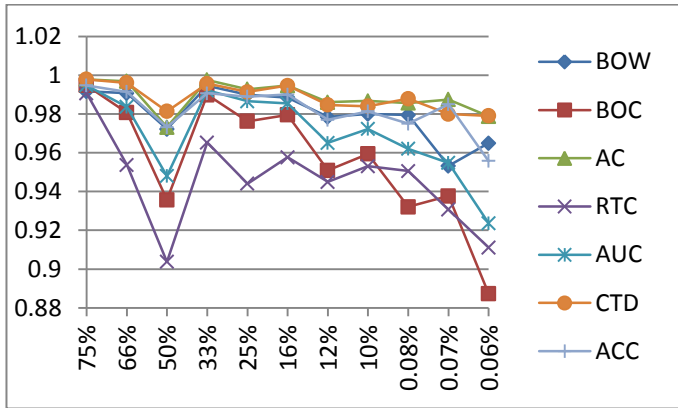


Figure 4. $F1$ measure on Classic3 Data set

Figure 4 shows a performance comparison, in terms of out two combined models based classification system of the BOW and BOC based approaches, when applied to Classic 3 dataset . In this experiment, the following percentages of the dataset have been used for the training, with the remainders utilised for testing (75%, 66%, 50%, 33%, 25%, 16%, 12%, 10%, 0.08%, 0.07%, 0.06%). Our combined model based system shows improvement compared to those achieved by the BOW and BOC vector methods individually.

To complement above results, Table VII shows the number of features utilised in each of the seven models compared above, indicates the complexity of the document representation vector used in each.

Table VII. Number of features in each model in classic 3 Dataset (3/4)

The scheme	Number of features
BOW	6446
BOC	11250
AC	16860
RTC	6446
AUC	13743
CTD	6446
ACC	11631

Figure 5, shows the experimental performance on 20 Newsgroups dataset which support the evidence previously discussed in Figure 4. It also shows that CTD achieves a

higher performance compared to other approaches and has a slight underperformance compared to the ACC [8].

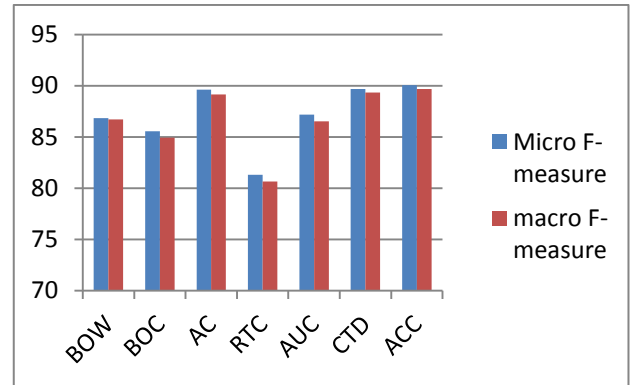


Figure 5. $F1$ measure on 20 Newsgroups Data set

Table VIII shows the number of features created using each of the above approaches for creating the combined model and how it affects the complexity of that representation.

Table VIII. Number of features in each model in 20 Newsgroups

The scheme	Number of features
BOW	40988
BOC	53206
AC	94194
RTC	40161
AUC	44797
CTD	40988
ACC	200126

V. CONCLUSION

In this work we have demonstrate that text classification can be improved by enriching documents with their related concepts using different approaches.

By doing so, background knowledge can be introduced into documents from sources such as Wikipedia to overcome some of the limitations of the BOW representation. We have described five different models for text representation which combine and enrich the BOW and BOC models.

The experiment result demonstrated the efficiency of four different capabilities of these models when applied to text classification, as compared to the BOW alone.

Among theses the CTD Model, which refers to "Using Concepts for Terms which do not appear in the Document" has been shown to achieve higher categorization accuracy while utilising the minimum number of features.

REFERENCES

- [1] Harris, Z.: Distributional structure. *Word* 10 (23), 146–162 (1954).
- [2] G. Miller. WordNet: A lexical database for English. *CACM*, vol. 38(11):pp. 39–41, 1995.
- [3] <http://www.wikipedia.org/>
- [4] A. Hotho, S. Staab, and G. Stummee. Wordnet improves text document clustering. *SIGIR :Workshop on Semantic Web*, 2003.
- [5] E. Gabrilovich, S. Markovitch, Feature generation for text categorization using world knowledge. In: *Int. Joint Conference on A.I IJCAI*, pp. 1048-498, 2005.
- [6] E. Gabrilovich, S. Markovitch. Overcoming the brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge. In: *Proc of the 21nd AAAI Conf on artificial intelligence AAAI'06*, pp. 1301-1306, 2006.
- [7] Hu, J., Fang, L., Cao, Y., Zeng, H., Li, H., Yang, Q., & Chen, Z. Enhancing text clustering by leveraging wikipedia semantics. In *Proceedings of the 31st ACM SIGIR*, pp. 179–186, 2008.
- [8] P.Wang, J.Hu, H-J. Zeng, Z. Chen. Using Wikipedia knowledge to improve text classification. *Knowledge and Information Systems*, vol. 19:pp. 265–281, 2009.
- [9] D. Milne & I.H. Witten. An open-source toolkit for mining Wikipedia, in *Proc. New Zealand Computer Science Research Student Conf*, Vol. 9.2009.
- [10] K. Lang. Newsweeder: learning to filter netnews. In: *Proc of the 12th Int Conf on machine learning ICML*, pp. 331–3393, 1995.
- [11] C. j. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, vol. 2(2), pp. 121–167, 1998.
- [12] J. Yun, L. Jing, J. Yu, H. Huang. A multi-layer text classification framework based on two-level representation model. *Expert Systems with Applications*, Vol. 39, p. 2035–2046, 2012.
- [13] Ricardo A. Baeza-Yates, Berthier Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1999.
- [14] G. Salton, A. Wong, & C.-S. Yang. A vector space model for automatic indexing'. *CACM*, vol. 18(11), pp. 613-620, 1975.